

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Институт машиностроения

(наименование института полностью)

Кафедра

«Промышленная электроника»

(наименование)

13.03.02 Электроэнергетика и электротехника

(код и наименование направления подготовки / специальности)

Интеллектуальные энергетические системы

(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Учебный стенд трех осевого манипулятора на ПЛК Siemens

Обучающийся

В. А. Новиков

(Инициалы Фамилия)

(личная подпись)

Руководитель

к.т.н., доцент, А. В. Прядилов

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Консультант

к.п.н., доцент, С.А. Гудкова

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Аннотация

Выпускная квалификационная работа посвящена созданию станда и практическому внедрению автоматизации трех осевого станка с помощью ПЛК Siemens с использованием LAD и SCL блоков программирования. Преимуществом данного станда является открытая кабельная конфигурация, технически и эргономически верный блок управления, который сопрягает станд Fischertechnik с ПЛК через шлейф, а также набор лабораторных работ с различными вариациями заданий для постепенного освоения автоматизации станда.

В данной работе содержатся аналитические алгоритмы, блок схемы и программы LAD и SCL которые имеют унифицированные переменные для удобства понимания алгоритма работы станда.

Данная работа содержит введение, теоретический раздел, практический раздел, набор лабораторных работ, заключение и список используемой литературы. Пояснительная записка изложена на 78 страницах. В ней имеется 6 таблиц и 61 рисунок. Графическая часть содержит 8 листов формата А1.

Abstract

The title of the graduate qualification work is «The training stand of the three-axis manipulator on the PLC Siemens».

The senior paper is devoted to the creation of a test bench and practical implementation of automation of a three-axis machine tool with a Siemens programmable logic controller (PLC) using ladder diagram (LAD) and Structured Control Language (SCL) programming blocks. The advantage of this test bench is the open cable configuration, the technically and ergonomically correct control unit, which interfaces the Fischertechnik test bench to the PLC via a loop, and a set of laboratory works with different variations of tasks for the gradual mastering of the test bench automation.

This work contains analytical algorithms, block diagrams, LAD, and SCL programs that have unified variables for easy understanding of the bench algorithm.

This work contains an introduction, a theoretical section, a practical section, a set of laboratory works, a conclusion and a list of references. The explanatory note is laid out on 78 pages. It has 6 tables and 61 figures. Graphic part contains 8 sheets of format A1.

Оглавление

Введение	5
1. Теоретическая часть.....	6
1.1 Стенд.....	6
1.2 Локационное устройство стенда	8
1.3 ПЛК и периферия.....	9
1.3.1 ПЛК Siemens	9
1.3.2 Блоки питания.....	11
1.4 Задача направления приводов	14
1.4.1 Включение в схему релейного управления	14
1.4.2 Энкодеры и начальная точка движения	15
1.5 Обратная связь	16
1.5.1 Оптопары.....	16
1.6 Анализ стенда в сборе.....	19
1.6.1 Принципиальная схема стенда	20
1.6.2 Шлейф.....	21
1.7 Корпус блока управления	24
2. Практическая часть	28
2.1 Пусконаладочный процесс	28
2.1.1 Аппаратный процесс	28
2.1.2 Программный процесс	34
2.2 Программа автоматизированного управления стендом.	38
2.2.1 Алгоритм работы.....	38
2.2.2 LAD и SCL программы	48
3. Лабораторные работы	64
3.1 Лабораторная работа №1 «Ручное управление стендом»	64
3.2 Лабораторная работа №2 «Работа с высокоскоростным счетчиком стенда»	66
3.3 Лабораторная работа №3 «Автоматизированное движение»	68
3.4 Лабораторная работа №4 «Создание циклического автоматизированного движения»	70
3.5 Лабораторная работа №5 «Полно структурный проект TiaPortal»	72
Заключение.....	74
Список используемой литературы.....	75

Введение

В настоящее время основы автоматизации вводятся во всех учебных заведениях с техническими направлениями такими как, промышленная электроника, наноэлектроника, электротехника и подобными, таким образом, чтобы усвоение базы электроники и микроэлектроники шло наравне с их интеллектуальным управлением. Именно поэтому автоматизация простых приводных станков является актуальной темой. Но к сожалению учебные станки и стенды от производителей ПЛК являются дорогостоящей аппаратурой, поэтому встает проблема поиска дешевого электростанка подходящего для контроллера, таким образом, чтобы станок отвечал требованиям и не уступал по функционалу настоящему станку, а будущий стенд состоящий из системы управления и станка развивал полноценные навыки программирования реального станка или локальной системы. Проект сопряжения такого конструктора как Fischertechnik, является показательным примером введения и совмещения микроэлектроники с возможностью автоматизации с относительно низкими финансовыми затратами. Безусловно, сам конструктор поставляется, в большинстве случаев, с входящим в комплект контроллером «ROBOTICS TXT», но данный программируемый контроллер в большей степени отвечает требованиям студентов-программистов, использующих такие языки программирования как C, C++, C# и VB, что повышает порог вхождения в среду автоматизации, а также смещает фокус на сами языки программирования, вместо совмещения программирования с электроникой. В данном случае для внедрения автоматизации в сферу электроники, следует использовать понятные студенту по примеру принципиальной электрической схемы, языки программирования использующие блочные способы отображения прохождения сигнала от «входа» к «выходу», такие как LAD и SCL, которые обширно используются в среде программирования ПЛК Siemens - Step7.

1. Теоретическая часть

1.1 Стенд

Предоставленный станок является трех-осевым манипулятором, который имеет три электродвигателя постоянного тока, два из которых имеют энкодеры- датчики угла поворота, которые выдают цифровой сигнал, который в дальнейшей программе является системой обратной связи с контроллером, для регулировки положения манипулятора.

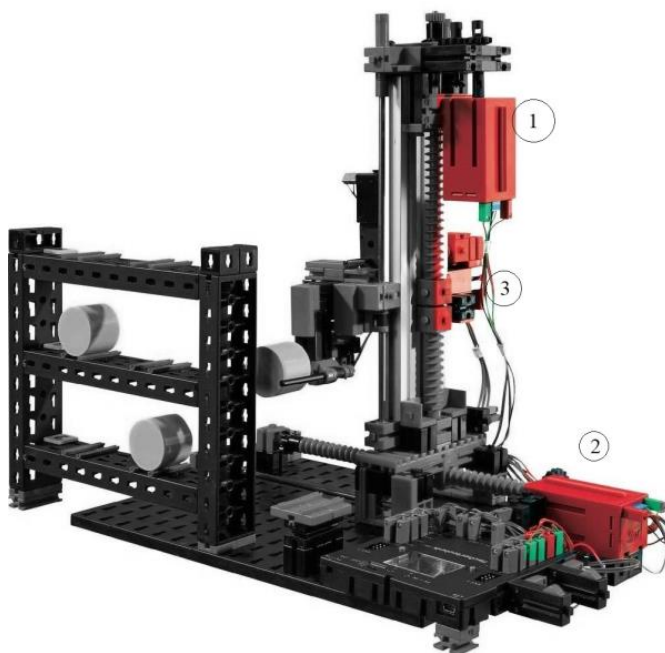


Рисунок 1 – Электродвигатели постоянного тока номер 1,2,3
соответственно

Электродвигатели номер один и два снабжены энкодерами (энкодер - это устройство, которое преобразует информацию из одного формата в другой. Например, энкодер может превращать движение валов в цифровой сигнал, который компьютер может обрабатывать), электродвигатель номер три имеет обратную связь с ПЛК за счёт кнопок концевиков (концевой выключатель - это устройство, которое используется в электрических цепях для определения механического положения или конца движения объекта, отображающих для программы начальное и конечное положение). Также стенд сохраняет за собой

возможность модификации в виде замены электродвигателя номер три на соответствующий двум другим электродвигатель с имеющимся в конфигурации энкодером, но базовая комплектация конструктора не подразумевает наличие третьего электродвигателя с энкодером.

Вид данного отслеживания положения манипулятора, обусловлен приближенностью данного аспекта отслеживания к реальному, так как на полномасштабных станках не используются такие системы отсчета как временная, замыкающая контакты. Так как данные виды отслеживания имеют множество недостатков по сравнению с внутренним предусмотренным в данном случае датчиком угла поворота. Временная система работы приводов может давать сбои по причине снижения напряжения питающей сети, соответственно будет снижаться и скорость вращения, и манипулятор не будет достигать предположенной программой точки, а система концевиков определяющая положения манипулятора имеет также ряд недостатков, таких как требование для каждого концевика-замыкателя отдельного входа, физический износ, наличие свободного хода, а также износ системы перемещения, потому как электродвигатель с манипуляционной системой имеют массу, в следствии чего появляется инерция, которая продолжает движения манипулятора по траектории упора, что не предусмотрено станком, и чрезмерно изнашивает его. Система счета поворотов двигателя нивелирует две обширные проблемы: первой – не зависит от времени и количества работающих электродвигателей – положение указано в системе отсчета поворотов по трем осям относительно нулевой точки координат, второй – в расчет точек движения включается момент инерции и дополнительное движение электродвигателей и манипулятора, путем внесения дополнительных счетных импульсов в ячейку данных координаты остановки манипулятора.

1.2 Локационное устройство стенда

Среда движения представляет собой поле из шести положений нахождения фишки, одно из которых всегда свободно для передвижения фишек по полю.

Реальный вид рабочего поля приведён на рисунке 2



Рисунок 2 – Реальный вид рабочего поля

Система движения имеет шесть координационных точек в двух осевом представлении, три вертикальных координаты и две горизонтальных координаты. Для упрощения представления требуется обратиться к рисунку 3. «Более сильной математической структурой является поле. Поле - это множество, которое может быть использовано для выполнения операций сложения, вычитания, умножения и деления. Если группа может только складывать и вычитать элементы, то кольцо может выполнять все эти операции, а поле также может делить элементы. [12, с 6]»

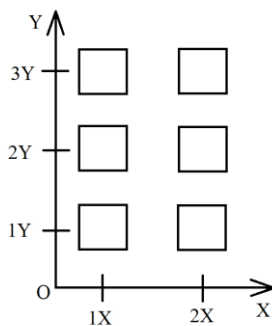


Рисунок 3 – Схематичное представление координат движения манипулятора

Прямоугольниками обозначены координационные места помещения фишек, а константы «X» и «Y» являются координатами, представленными в данном случае в условном виде, но в программе каждая ордината и абсцисса представлена в виде целочисленного числа, полученного путем счета импульсов поворотного датчика с инерционным учетом.

Абдуллин Э.Б. в своей книге «Автоматизация координатных измерений в машиностроении» писал, что для вычисления координат теоретической точки на плоскости через блок данных, необходимо записать координаты трех измеренных точек в блок. Затем можно вызвать внешнюю программу для расчета и сохранения результатов в файл. После этого, можно прочитать результаты из файла и вывести их на окончательный отчет [1, с 45].

Третья ось не является полноценно отслеживаемой осью движения так как она имеет всего два положения отслеживающихся концевыми выключателями, положение «максимально вытянутого манипулятора» и положение «минимально вытянутого манипулятора». Данные положения обеспечивают подъем и сброс фишки на координационных местах приема и выдачи, и не нуждаются в отдельной ячейке данных для расчета. Но возможность модификации для точного принятия и сброса фишки, конфигурацией программы и модели сохраняется.

1.3 ПЛК и периферия

1.3.1 ПЛК Siemens

Моделью используемого ПЛК является Siemens S7-1200 1215DC.

Как описывает С.Н. Басков, «Аппаратно ПЛК является вычислительной машиной, поэтому архитектура его процессорного ядра практически не отличается от архитектуры компьютера. Отличия заключены в составе периферийного оборудования, отсутствуют видеоплата, средства ручного ввода и дисковая подсистема. Вместо них ПЛК имеет блоки входов и выходов. [6, с. 3]»

Данный программируемый логический контроллер имеет 15 дискретных входных выводов и 10 дискретных выходных выводов. Блоком питания для контроллера является импульсный блок питания «ОВЕН БП60Б-Д4-24» с выходным напряжением 24 В и мощностью 60 Вт. Одной из главных особенностей данного ПЛК является отдельное параллельное питание 24В для ряда входных и выходных выводов, которое реализуется подключение в точки вывода «L+» для входного реестра и «4L+» для выходного, питания +24В, а также заземления в точки вывода «М» для входного реестра и «4М» для выходного. Это необходимо для считывания контроллером замыкания и размыкания контакта входных и выходных выводов. Наглядно данные выводы можно увидеть на рисунке 4.

«Развитие вычислительной техники привело к появлению микропроцессорных автоматических систем (МПАС) управления, в которых контур управления замыкается только на определенные промежутки времени, воздействуя импульсами на исполнительный механизм [8, с.16]»

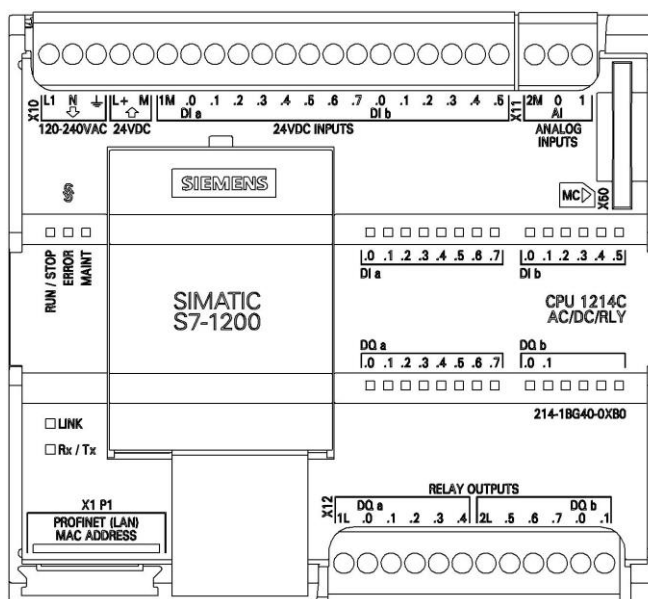


Рисунок 4 – Конфигурация Siemens S7-1200

Аналоговые выводы, а также выводы переменного тока в данном проекте не используются, по причине дискретно-булевых операций с номинальным током 4.20 мА.

Отсюда можно сделать вывод, что для работы стенда требуется два блока питания 24В для контроллера и 9В для системы приводов конструктора.

1.3.2 Блоки питания

Выбор блока питания 9В проходил по характеристикам мощности, стоимости, и количеству выводов выходного постоянного напряжения.

Первоначальной точкой сравнения был выбран блок питания Fischertechnik Power Set, производитель Германия <https://расрас.ru/product/505283/> (рисунок 5).

Набор с блоком питания



7650.00 руб.

Артикул: 505283
Вес: 0,65 кг
Страна производителя: Германия
Размер упаковки (ШхВхГ): 22,5х6,5х15 см
Оригинальное название: Power Set

Кол-во: [В корзину](#)

Сетевой блок питания для конструкторских наборов fischertechnik. Кроме блока питания в состав набора входит регулятор напряжения, обеспечивающий диапазон регулирования 0-9 В, и разъемы для подключения моделей.

Напряжение на выходе регулятора устанавливается с помощью вращающейся ручки. Также имеется переключатель полярности.

Состав набора:

- Блок питания: 1 шт.
- Регулятор напряжения: 1 шт.

Характеристики блока питания:

- Входное напряжение: переменное 220 В
- Выходное напряжение: 9 В
- Максимальный выходной ток: 2,5 А
- Диаметр цилиндрического разъема со стороны 9 В: 3,45 мм

Характеристики регулятора напряжения:

- Входное напряжение: постоянное 9 В
- Выходное напряжение: регулируемое 0-9 В
- Максимальный выходной ток: 1,0 А
- Переключатель полярности регулируемого выхода
- Дополнительный нерегулируемый выход: 9 В, 1 А

Рисунок 5 – Блок питания Fischertechnik Power Set

По заявлению производителя данный блок питания рассчитан на 22,5Вт, а также после блока регуляции мощность снижается до 9Вт на каждый из двух выходов. Значительным недостатком данной модели является мощность, потому как отечественный блок питания ОВЕН БП60Б-Д4-9 (рисунок 6) <https://www.owenkomplekt.ru/odnokanalnyj-blok-pitaniya-oven-bp60b-d4-9.html> имеет всего один выходной канал, однако предоставляемой мощностью является 60Вт, иначе говоря 6,67А что в 6,67 раз больше по сравнению с Fischertechnik Power Set. Преимущества такие как регулятор напряжения также не требуются в конфигурации стенда, так как снижение напряжения сети будет приводить к снижению скорости перемещения манипулятора, что в рамках технического обучения является рудиментарным. Также отечественный блок питания имеет ниже стоимость, а также включает в себя

возможность ремонта, из-за удобной внутренней конфигурации. Стоит отметить что ОВЕН также имеет возможность регулировки выходного напряжения, но по сравнению с германским аналогом возможность случайного изменения выходного напряжения случайным касанием заранее предотвращена, винтовой глубоко посаженной регулировкой (вывод «РЕГУЛИР»).

Артикул: 00-00031290

Категории:	Блоки питания и устройства коммутации
Выходное напряжение:	9 В
Количество каналов:	1
Выходная мощность:	60 Вт
Входное напряжение:	90...264 В
Вид монтажа:	на DIN-рейку
Вид напряжения:	от 220 В AC

Блок питания БП60 предназначен для питания стабилизированным напряжением постоянного тока различных широкого спектра радиоэлектронных устройств (релейной автоматики, контроллеров, датчиков и т.п.).

Подобрать описание >

6 480 руб. под заказ

Заказать + 1 - ШТУК

Добавить в корзину

[К сравнению](#) [Экспресс-заказ](#)

Информация О ДОСТАВКЕ [По Москве](#) [В регионы](#)

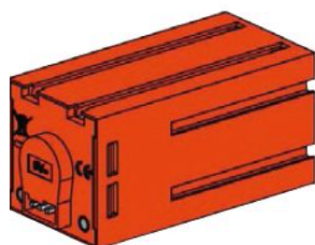
Информация ОБ ОПЛАТЕ [Подробнее](#)

Рисунок 6 – Блок питания ОВЕН БП60Б-Д4-9

Важность мощности в аргументах выбора блока питания приведена по причине анализа характеристик электропривода станда.

Стандартным электродвигателем постоянного тока с энкодером в станде является Encodermotor 9V neu (рисунок 7). <https://расрас.ru/product/153422-motor-s-enskoderom-novij/>

Мотор с энкодером 9 В (2014)



3570.00 руб.

Артикул: 153422
 Вес: 0,105 кг
 Страна производителя: Германия
 Оригинальное название: Encodermotor 9V neu

Кол-во: [В корзину](#)

Мотор-редуктор со встроенным энкодером.

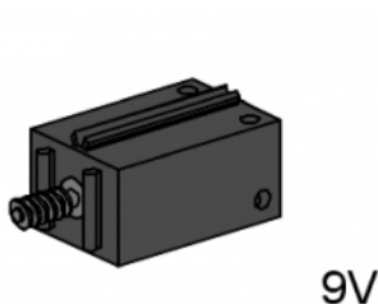
Характеристики:

- Тип электродвигателя: коллекторный двигатель постоянного тока
- Тип возбуждения: от постоянных магнитов
- Напряжение питания: 9 В
- Скорость вращения: 105 об/мин
- Номинальный ток: 0,386 А
- Номинальный момент: 10,9 Н*см
- Передаточное число редуктора: 21,1:1
- Энкодер: импульсный - 63,3 имп/об
- Тип выхода энкодера: открытый коллектор NPN

Рисунок 7 – Encodermotor 9V neu

Суммарно для трех моторов требуется 1,158А или 10,5Вт, а в конфигурации два с энкодером и один без энкодера (рисунок 8) 1,037А или 9,3Вт, что в любом случае требует обеспечения питанием больше 9Вт с одного выхода. Соответственно блок питания ОВЕН обеспечивает с большим запасом всю систему, а также сохраняет потенциал к доработке и модернизации стенда для более сложных конфигураций.

Мотор XS 9 В



870.00 руб.

Артикул: 137096
Вес: 0.0016 кг
Страна производителя: Германия
Оригинальное название: MOTOR XS KOMPL. MONT.

Кол-во:

 **В корзину**

Миниатюрный мотор. Напряжение питания 9 В.

Характеристики:

- Напряжение питания: 9 В
- Скорость вращения: 5995 об/мин
- Номинальный ток: 0,265 А
- Номинальный момент: 0,15 Н*см

Рисунок 8 – Электромотор постоянного тока без энкодера

Работа моторов, а также задание направления происходит за счет двух реле на один мотор, линия коммутации которых проходит через выходные выводы ПЛК, а силовая линия проходит через блок питания 9В соответственно. Реле в своей сущности представляет собой коммутатор с двумя положениями NC («нормально закрытым») и NO («нормально открытым»), подача импульса на управляющий вывод коммутирует переход из NC в NO. «Контакт NO или NC используется для сканирования состояния сигнала двоичной метки. При отсутствии контакта состояние отсканированного сигнала передается непосредственно в логическую операцию[26, с. 2]»

1.4 Задача направления приводов

1.4.1 Включение в схему релейного управления

Принцип задачи направления вращения вала мотора заключается в способе включения его в сеть, иначе говоря подачи напряжения на стандартное включение прямого хода или обратное включение – реверсивное (рисунок 9). «Напряжение U характеризует собой энергетические возможности поля в данной области пространства. Электрическое напряжение является скалярной величиной и в общем случае может приобретать как положительные, так и отрицательные значения» [2, с. 12] Данный процесс называется – реверс электродвигателя.

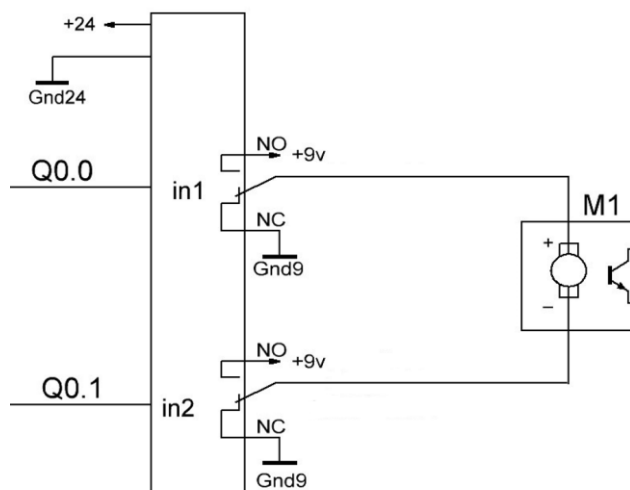


Рисунок 9 – Пример включения электромотора с задачей направления от контроллера

Первоначально, как показано на схеме оба вывода реле находятся в положение NC, соответственно мотор имеет две одинаковые полярности напряжения, что говорит об отсутствии протекания тока по обмотке мотора соответственно движения привода не происходит. В моменте выполнения алгоритма на один из управляющих выводов двух реле приходит сигнал, и одна из реле переводит положение силового вывода из NC в NO, соответственно задает вывод с большим потенциалом что ведет за собой протекание тока в заданном направлении и происходит вращения привода.

Преимуществом данного соединения является отсутствие запрещенного режима, что является безопасным для студента-пользователя в момент работы с программой. Также безопасность работы станка обеспечивает прерывание работы отсутствием импульса с ПЛК на управляющий контакт реле, что упрощает аварийное отключение в случае ошибки в программе, а также работы в упор. Стоит отметить, станд следует отключать от сети в «нулевом» положении или имея заранее подготовленную программу движения в нулевое положения, во избежание движения манипулятора в упорные положения.

«Положение координатной системы станка определяется нулевой точкой – ноль станка.[21, с. 56]»

1.4.2 Энкодеры и начальная точка движения

Два электромотора станда обособляются энкодерами на 63,5 импульса за период. И скоростью вращения 105 оборотов в минуту. Соответственно, за одну секунду происходит 1,75 поворотов, что равняется примерно 112 импульсам в секунду с энкодера, или 112 Гц. Частота дискретизации входных портов по заявлению производителя равняется 30 кГц без аппаратного включения скоростного счетчика. Однако практика показывает, что назначение аппаратного счетчика на входы, задействованные в обратной связи энкодеров, снижает неподсчитанные импульсы вплоть до нуля, по сравнению с базовым включением без аппаратного счетчика «High speed counter».

Одной из немаловажных частей функционирования является начальная точка движения манипулятора. Потому как электромотор конструктора Fischertechnik оснащен полузамкнутым контуром регулирования, и в свою очередь, как пишет в своей книге «Следящие электроприводы с ЧПУ» Лебедев А.М. «<...> датчик обратной связи по положению (револьвер или импульсно-кодовый датчик) установлен на валу двигателя и измеряет угол поворота его вала, механическая часть станка не охвачена обратной связью, поэтому точность позиционирования станка в основном зависит только от механической системы. [16, с. 126]» Соответственно, любой пуск

манипулятора для совершения своей функциональной задачи, необходимо производить исключительно из начальной точки, потому как в начальный момент работы станка энкодер может быть замкнут, и посылать в начальный момент времени в ячейку хранения импульсов, значение о вращении, что при отсутствие обнуления может привести к рассинхронизация физического положения с математическим расчетом и дальнейшим нежелательным последствиям. Постановка нулевого положения, задается тремя концевиками, при нажатии одновременно на которые, происходит обнуление счетных ячеек данных. Данная подсистема обнуления, или начального положения, является одной из базовых подпрограмм необходимых для начала программирования основной функции станка, и стоит наравне с такими подсистемами как кнопка аварийной остановки и остановки всех расчетов в системе, в противном случае оператор будет вынужден обесточить стенд, что является неблагоприятной процедурой для системы.

1.5 Обратная связь

1.5.1 Оптопары

Процесс согласования напряжения обратной связи с приводами, происходит с помощью оптопар. Оптопара является четырехвыводным устройством, две пары из которого принадлежат входной цепи - излучателю, и две выходной цепи - фотоприемнику. Для данной конфигурации стенда были подобраны быстродействующие оптопары фирмы TOSHIBA модели tlp627-1 (рисунок 10).

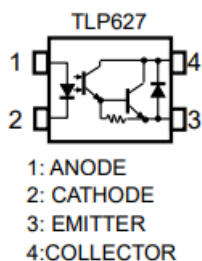


Рисунок 10 –Принципиальная схема устройства оптопары

Данная оптопара как видно по ее внутренней конфигурации является транзисторной соответственно ее функционирование происходит следующим образом. При протекании по цепи излучателя анод-катод номинального тока возникает фототок, в тоже время являющимся базовым для транзистора выходной цепи, при номинальном поступлении тока, в данном случае с транзисторного энкодера, ток будет достаточен для отпирания транзистора. Однако стоит ограничить входной ток резистором номиналом 510 Ом, так как мощность излучателя составляет 100 мВт, что в пересчете без дополнительной нагрузки дает 11мА, что превышает максимальный ток 10мА и номинальный 8.3мА. По закону Ома получается, что сопротивление оптопары $1.2/0,083=144,6$ Ом Также стоит отметить, что напряжение прямого включения составляет 1.15В номинальное и 1.3В максимальное (рисунок 11). Соответственно требовалось снизить падение напряжение и ток цепи до номинального, поэтому было выбрано сопротивление, у которого сопротивление в 6 раз больше сопротивления входного контура, и равняется 1кОм.

TOSHIBA

TLP627, TLP627-2, TLP627-4

Electrical Characteristics (Ta=25°C)

Characteristics		Symbol	Test Condition	Min	Typ.	Max	Unit
LED	Forward Voltage	V_F	$I_F = 10 \text{ mA}$	1.0	1.15	1.3	V
	Reverse Current	I_R	$V_R = 5 \text{ V}$	—	—	10	μA
	Capacitance	C_T	$V = 0 \text{ V}, f = 1 \text{ MHz}$	—	30	—	pF
Detector	Collector-Emitter Breakdown Voltage	$V_{(BR)CEO}$	$I_C = 0.1 \text{ mA}$	300	—	—	V
	Emitter-Collector Breakdown Voltage	$V_{(BR)ECO}$	$I_E = 0.1 \text{ mA}$	0.3	—	—	V
	Collector Dark Current	I_{CEO}	$V_{CE} = 200 \text{ V}$	—	10	200	nA
			$V_{CE} = 200 \text{ V}, T_a = 85 \text{ }^\circ\text{C}$	—	—	20	μA
Capacitance Collector to Emitter	C_{CE}	$V = 0 \text{ V}, f = 1 \text{ MHz}$	—	10	—	pF	

Рисунок 11 –Электрические характеристики оптопары

Одной из главных характеристик выбранной оптопары является быстроедействие. Суммарное время одного периода оптопары составляет 120 микросекунд (рисунок 12), что в пересчете на величину Герц равно 8.3КГц, а

максимальная частота системы составляет 120Гц, что удовлетворяет выбор по характеристике быстродействия. «На исполнительном уровне ИПС важнейшей подсистемой является АСД - автоматизированная система диагностики, которая имеет распределенную сеть датчиков, позволяющую анализировать состояние и нормальное функционирование ГПС (глобальной производственной системы) на всех уровнях производственного процесса и управления. [13, с. 10]»

TOSHIBA

TLP627, TLP627-2, TLP627-4

Switching Characteristics (Ta=25°C)

Characteristics	Symbol	Test Condition	Min	Typ.	Max	Unit
Rise Time	t _r	V _{CC} = 10 V I _C = 10 mA R _L = 100 Ω	—	40	—	μs
Fall Time	t _f		—	15	—	
Turn-on Time	t _{on}		—	50	—	
Turn-off Time	t _{off}	—	15	—		
Turn-on Time	t _{ON}	R _L = 180 Ω (Fig.1) V _{CC} = 10 V, I _F = 16 mA	—	5	—	
Strage Time	t _s		—	40	—	
Turn-off Time	t _{OFF}		—	80	—	

Рисунок 12 – Временные характеристики быстродействия оптопары

Напряжение на выходном контуре подается с блока питания 24В питающего логический контроллер, а также его входные выводы. Данного напряжения достаточно для насыщения коллектора выходного транзистора, что не требует дополнительного форсирования током измененной амплитуды, потому как конденсатору величиной 10пФ выходного контура достаточно времени перезарядится, и дополнительная оптимизация выходного контура не требуется.

На схеме стенда оптопара имеет следующее подключение к энкодеру (рисунок 13).

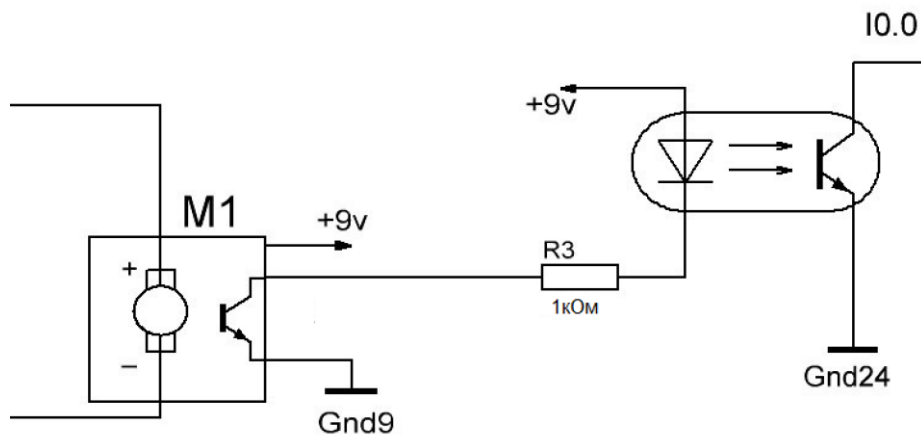
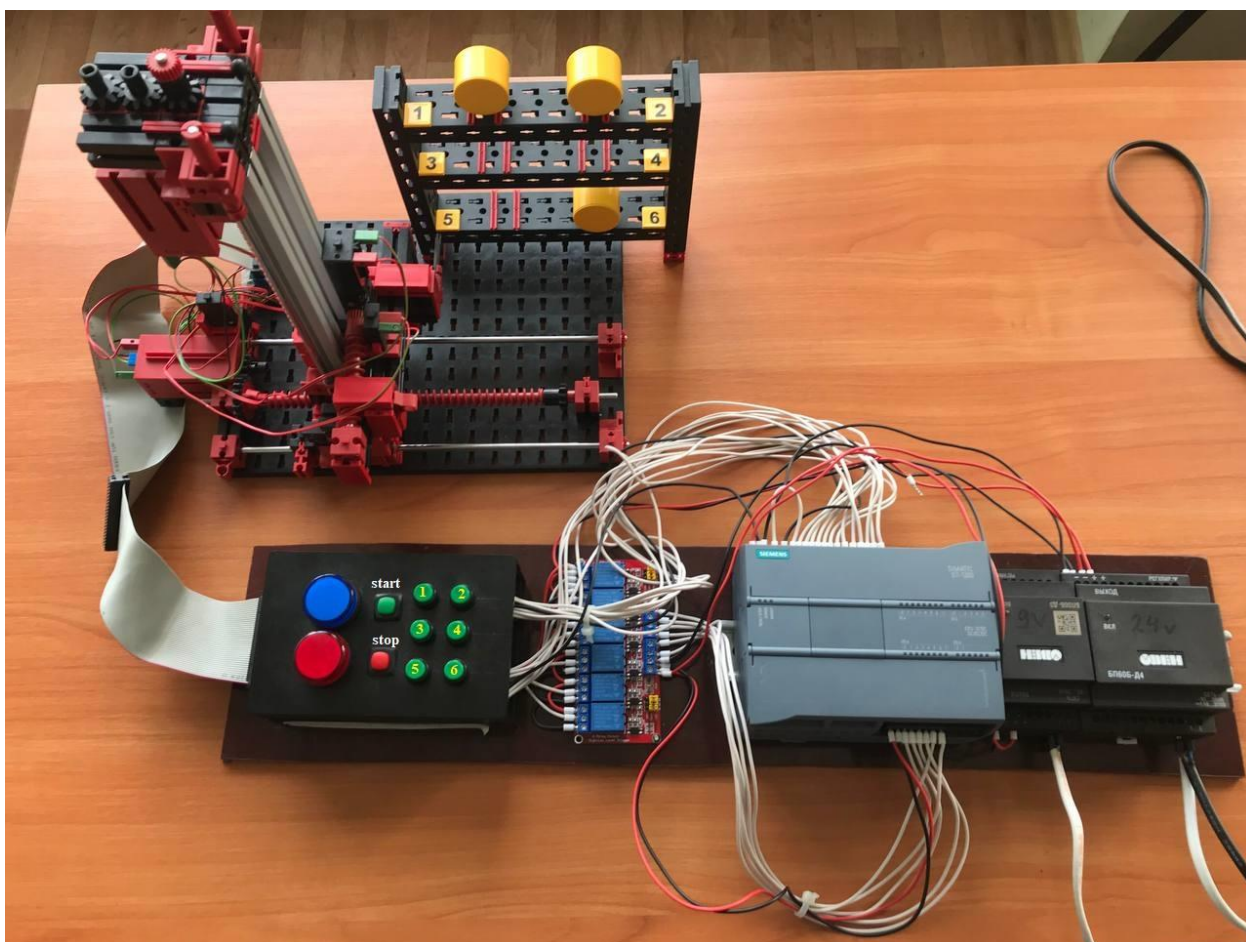


Рисунок 13 –Схема подключения оптопары к энкодеру

1.6 Анализ станда в сборе

Стенд манипуляционного станка с силовой и управляющей частью представлен на рисунке 14.



Рисунка 14 – Опытный образец станда манипулятора на базе Siemens

Рисунок 14 дает понимание общей составной конфигурации стенда, которую разделяет сорока пионовый шлейф в который входит питание 9В с заданным направлением от блока согласования реле, а также обратная связь с контроллером посредством оптопары, находящейся в блоке клавишного управления. «ПЛК должен быть смонтирован в вертикальном положении на ровной, не горючей поверхности в закрытой, электрической рабочей зоне. [20, с. 8]»

Важной частью проектирования является отделение силовой части, а именно блоков питания и идущим к ним проводам с переменным питанием от розетки, от линий обратной связи энкодеров, так как особенностью данной электроустановки является множество кабелей, на которые могут оказывать влияния наводки. Поэтому в данном случае размещение вдали от кабелей в которых протекает высокочастотный ток, кабелей переменного напряжения 50 Гц является решением, предотвращающим такие процессы как гармоники, вышеописанные 100-112 Герц обратной связи энкодеров практически равняются второй гармонике, что является основанием для предотвращения электромагнитных возмущений¹.

1.6.1 Принципиальная схема стенда

Принципиальная схема стенда представлена на рисунке 15.

¹ О раздельном размещении также рекомендует Рекус Г. Г. в «Общая электротехника и основы промышленной электроники»[18]

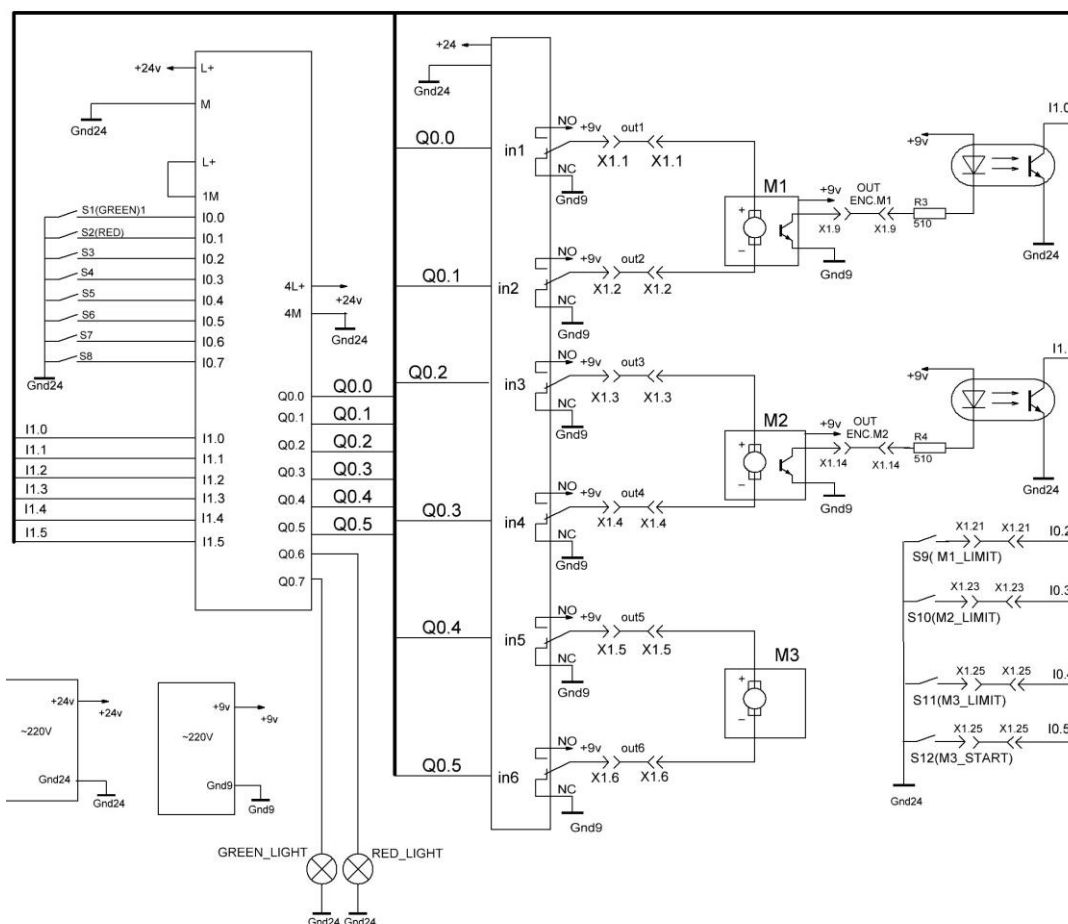


Рисунок 15 – Принципиальная схема стенда

1.6.2 Шлейф

Главной особенностью конфигурации стенда является наличие шлейфа обозначенным жирной линией на принципиальной схеме. Так как он обеспечивает долговечность работы стенда из-за дополнительного механического функционала защелкивания проводов, которые имеют особенность изнашиваться в процессе активного труда, а также предоставляет возможность отсоединения конструктора от блока силовой и управляющей части без дополнительных инструментов и запоминания присвоения выводов контроллера, что в процессе доработки или модернизации является достоинством стенда.

Игольчатые части шлейфа расположены со стороны блока клавишного управления и конструктора рисунок 16.



Рисунок 16 – Расположение игольчатого коннектора в блоке клавишного управления

По бокам узла расположены отверстия для фиксации вставного шлейфа, с верха располагается вырез под «ключ» вставного коннектора, определяющий ротацию вставного шлейфа. Схематичный пример вставного шлейфа расположен на рисунке 17.

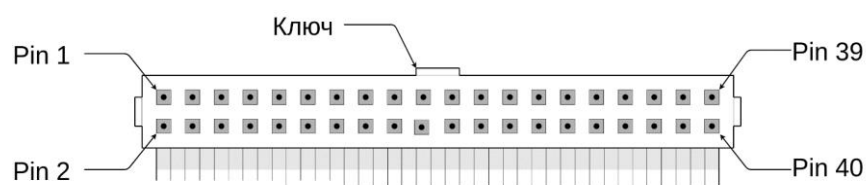


Рисунок 17 – Вставной сорока пиновый разъём

Для удобства подключения таких разъемов используются таблицы подключений, пример которой для данного стенда представлен в таблице 2.

Важно сохранять ротационное положение шлейфа, для подключения и назначения выводов, а также учитывать, что расположения карты выводов для

вставного и игольчатого коннектора зеркально отличаются. Для этого вдоль одной из частей шлейфа располагается красная линия как ориентир положения. Далее представлена таблица расшифровки названий выводов в таблице распиновки (Распиновка шлейфа - это определенная последовательность контактов на разъемах шлейфа, которая устанавливает соединение между устройствами. Каждый контакт имеет свой номер и назначение, например, передача данных, питания и т.д. Правильное подключение контактов по распиновке очень важно для правильной работы устройства, к которому подключен шлейф). (таблица 1).

Таблица 1 – Расшифровка названий выводов в таблице распиновки

Название тега	Назначение
+ M1 / - M1	Выводы для первого – горизонтального привода с реле
+ M2 / - M2	Выводы для второго – вертикального привода с реле
+ M3 / - M3	Выводы для третьего – диагонального привода с реле
+9V ENC.M1 / +9V ENC.M2	Питание +9В для энкодеров
GND9	Заземление 9В
OUT ENC.M1 I0.0	Информационный вывод привода энкодера M1
OUT ENC.M2 I0.1	Информационный вывод привода энкодера M2
GND24	Заземление 24В
I1.2 / I1.3 / I1.4 / I1.5	Информационный вывод ограничительных кнопок для приводов с выводами Di b I1.2 / I1.3 / I1.4 / I1.5 соответственно

Таблица 2 – Распиновка коннектора игольчатого 40 пин

Пин	Название в шлейфе	Подключение к Siemens	Пин	Название в шлейфе	Подключение к Siemens
2	-M1 (out2)	Q0.1	1	+M1 (out1)	Q0.0
4	-M2 (out4)	Q0.3	3	+M2 (out3)	Q0.2
6	-M3 (out6)	Q0.5	5	+M3 (out5)	Q0.4
8	GND9 ENC.M1		7	+9V ENC.M1	
10	OUT ENC.M1	I0.0	9	OUT ENC.M1	I1.0
12	GND9 ENC.M2		11	+9V ENC.M2	

14	OUT ENC.M2	I0.1	13	OUT ENC.M2	И1.1	
16			15			
18			17			
20			19			Ключ
22	GND24		21	И1.2 (M1_LIMIT)	И1.2	
24			23	И1.3 (M2_LIMIT)	И1.3	
26			25	И1.4 (M3_LIMIT1)	И1.4	
28			27	И1.5 (M3_LIMIT2)	И1.5	
30			29			
32			31			
34			33			
36			35			
38			37			
40			39			

1.7 Корпус блока управления

Корпус для блока клавишного управления (рисунок 18), включающий в себя игольчатый коннектор, был разработан специально для выполнения своей операторной функции с технологическими отверстиями под шесть кнопок положения, две кнопки начала и окончания процесса работы и две индикаторные лампочки работы и ожидания. Корпус разделен на две части под винт м4 для удобства монтажа на металлический профиль, а также внутреннего монтажа. Плотное прилегание и фиксация обеспечивают отсутствие дребезга, которое исключает изнашивание кабелей и непроизвольное замыкание открытых паяных или обжатых частей кабеля. С задней части расположены два технологических отверстия для проведения и соединения кабелей (рисунок 19)¹.

¹ Шах Анкит П. также описывает важность верного проектирования оболочки блоков управления в своей научной работе «Design Automation of Shell» («Автоматизация проектирования оболочки»)[27]

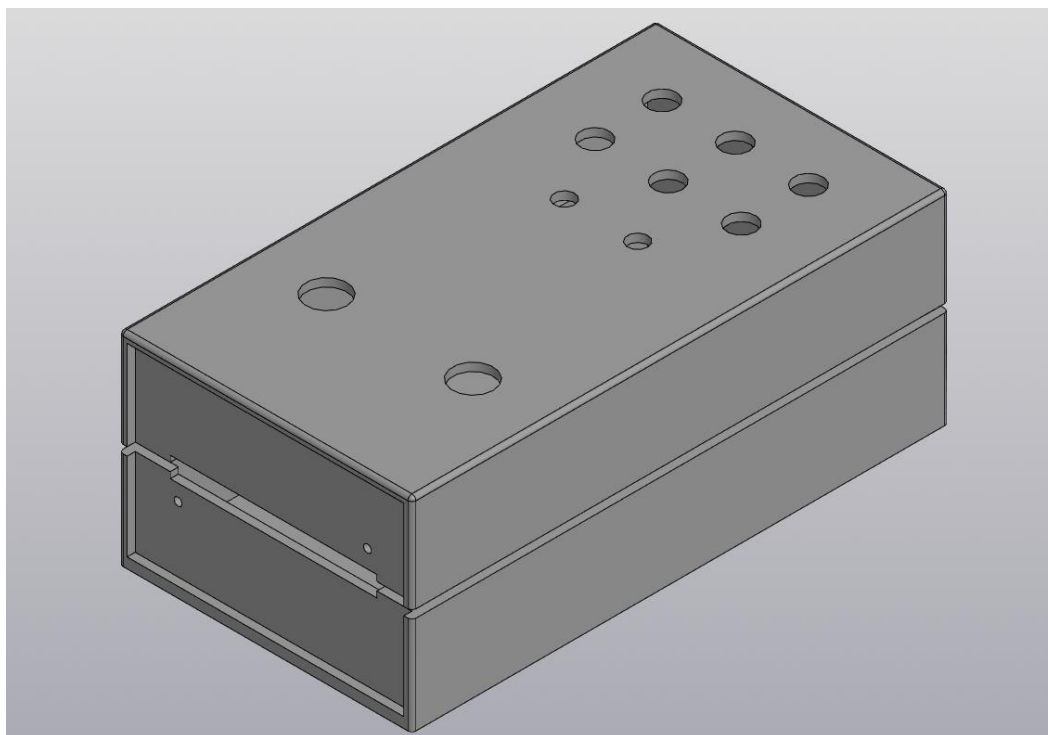


Рисунок 18 – Корпус для блока клавишного управления

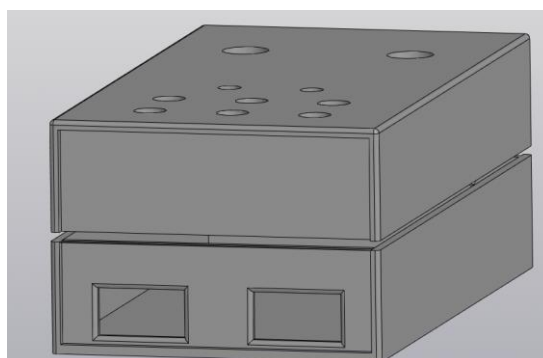


Рисунок 19 – Задняя часть с технологическими отверстиями для кабеля

Развязка проводов и блок управления не могли бы быть помещены в стандартную электромонтажную распределительную коробку (рисунок 20) по причинам не соответствия, делящихся на два типа конструктивные и эргономические.



Рисунок 20 – Стандартная электромонтажная распределительная коробка

Конструктивные: низкокачественный материал (мягкий прогибающийся пластик) распределительных коробок, распределенные отверстия по четырем сторонам, вместо двух, отсутствующая возможность встроить коннектор шлейфа абсолютно зафиксировано без лишнего повреждения грани корпуса, фиксация является фундаментальной характеристикой коммуникации посредством шлейфа, и одно из самых важных – нарушение модульности на металлическом профиле (DIN рейке) и нарушение совместимости монтажных панелей из-за субъектного сверления для крепежа что влечет за собой отсутствие унификации длины соединительных кабелей в случае замены одного блока клавишного управления другим.

«Представление формы в САПР. Кроме того, некоторые объекты подразумевают существование объектов «sketch», что подразумевает последовательность между 3D-объектами и «sketch». [22, с. 4]»

Существуют эргономические требования к органам управления, которые регламентируются по ГОСТ Р ИСО 1503-2014¹. Если не включать в учет общие требования технической эстетики к объектам проектирования, то монтажная распределительная коробка могла бы являться корпусом для блока клавишного управления, если бы конструкция крепежа верхней грани (крышки) фиксировалась на винты, а не защелки, что исключало бы случайные сдвиги и срабатывания.

¹ ГОСТ Р ИСО 1503-2014. Эргономика. Требования к пространственной ориентации и направлениям движения органов управления. [9]

В свою очередь, разработанный корпус сделан из поликарбоната одного из самых крепких полимеров, который имеет высокий класс устойчивости ГОСТ Р 51136 «Защитные стекла»¹. Черный цвет также был выбран с учетом эргономических требований ГОСТ, потому как цветовое кодирование блока управления не требуется. Естественным недостатком высокой твердости является, относительная хрупкость при ударах, но данный недостаток нивелируется креплением к DIN рейке, соответственно твердость как отсутствие деформации при стандартном усилии нажатия является достоинством. «6.3. Усилие нажатия одиночных клавиш должно быть от 0,35 до 1,00 Н, при этом значения в пределах одной клавиатуры не должны отличаться друг от друга более чем на 30%. [11, п. 6.3.]»

В данном разделе были рассмотрены такие темы как состав стенда и его принципиальное устройство, которое состоит из локационной плоскости координатного движения, рассмотрен контроллер и подобраны для стенда блоки питания, питающие ПЛК и релейный блок. Также применяя релейный блок, удалось добиться включения и назначения задачи направления приводов, управляемых дискретным выходом ПЛК. Рассмотрение обратной связи приводов через импульсные энкодеры и согласование напряжения обратной связи через оптопары, дало полносоставный стенд и возможность составить принципиальную схему, также был введен шлейф, который был интегрирован в корпус блока клавишного управления.

¹ ГОСТ Р 51136-2008. Стекла защитные многослойные. Общие технические условия [10]

2. Практическая часть

2.1 Пусконаладочный процесс

2.1.1 Аппаратный процесс

Пусконаладочный процесс стенда необходимо производить поэтапно, начиная от подключения силовой части к контроллеру, далее к блоку реле и к шлейфу. Целями пусконаладочной работы являются следующие пункты:

1) Проверка конфигурации системы на наличие короткого замыкания в процессе сборки и верной подачи напряжения, отображающейся зелеными светодиодами на частях блока управления (рисунок 21).



Рисунок 21 – Проверка индикаторов питания на блоках управления и силовой части

2) Проверка подачи питания на линию выходных и выходных линий ПЛК

3) Проверка контактов на входной – информационной линии блока реле, проверка контактов на выходной – коммутирующей линии блока реле (рисунок 22).

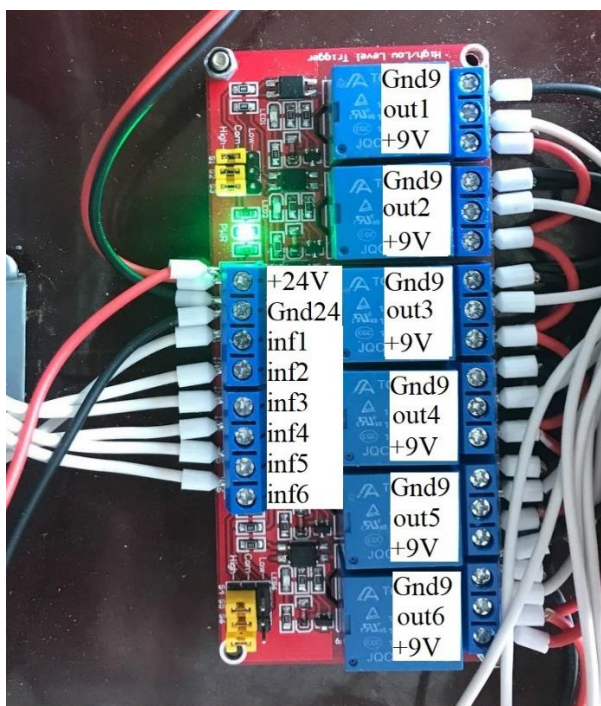
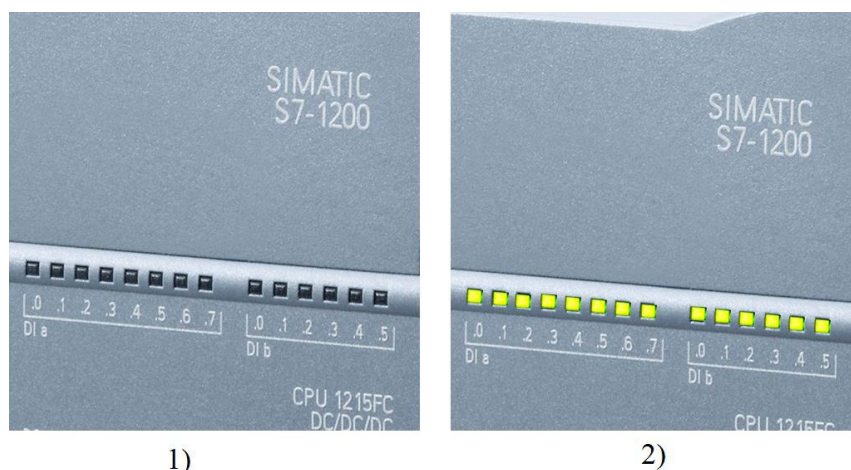


Рисунок 22 –Проверка подключения контактов к блоку реле

4) Проверка работы подключенных кнопок на индикаторном блоке ПЛК (рисунок 23).

«Некоторые средства тестирования могут применяться для поддержки автоматизации процесса или загрузки больших массивов данных из входных файлов. Нужно рассмотреть, какие задачи группа тестирования пытается решать с помощью инструмента тестирования.[7, с. 99]»



1)

2)

Рисунок 23 –Проверка коммутации входных выводов 1) ничего не коммутируется, 2) коммутируются все входные линии

5) Проверка кабеля подключения кабеля Ethernet (рисунок 24).



Рисунок 24 – Проверка индикатора LINK подключения кабеля Ethernet

Как описывает Остроух А.В, «для стандарта Ethernet была учтена необходимость назначения уникального шестибайтного MAC-адреса каждой сетевой карты и встроенного сетевого интерфейса для идентификации отправителя и получателя кадра. Производители уже закладывают этот номер при изготовлении устройств, что избавляет сетевого администратора от необходимости настраивать MAC-адрес при подключении новых устройств к сети.[15, с. 7]»

б) Программная проверка на пуско-наладочной программе работы электродвигателей (рисунок 25).

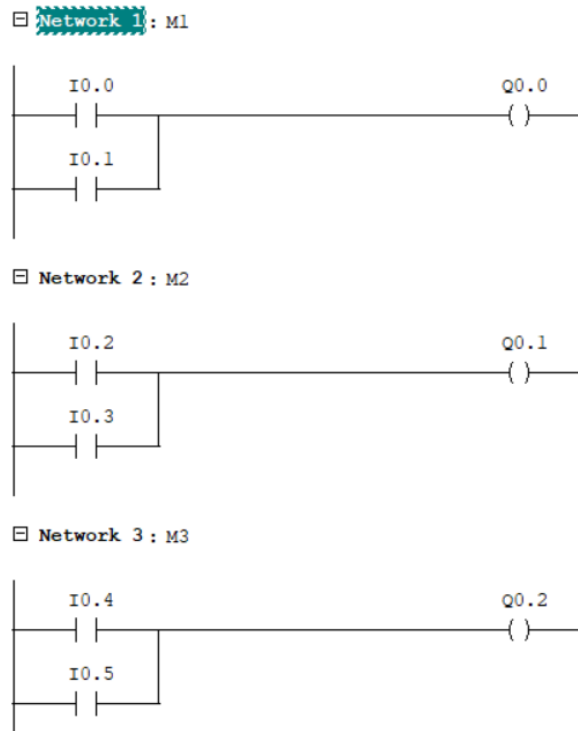


Рисунок 25 –Программа для проверки работы двигателей M1,M2,M3

7) Проверка обратной связи энкодеров с помощью пуско-наладочной программы (рисунок 26).

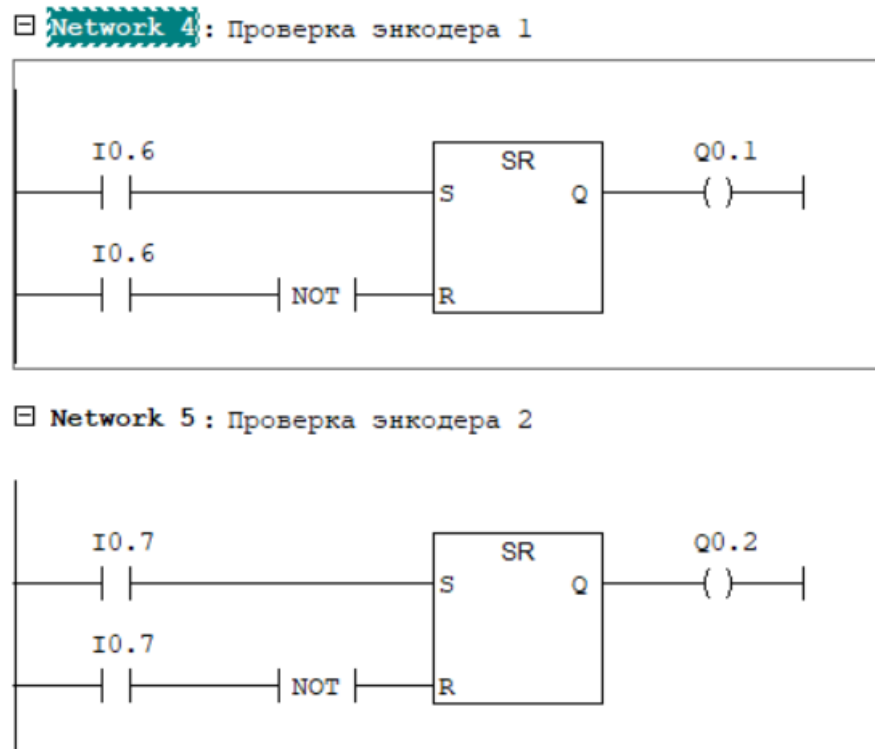


Рисунок 26 – Проверка обратной связи энкодеров двигателей M1, M2

8) Проверка нажимных концевиков крайнего положения по трем осям движения с помощью пуско-наладочной программы (рисунок 27).



Рисунок 27 – Проверка нажимных концевиков крайнего положения трех осей

Анализ шагов пусконаладочного процесса является крайне важной процедурой запуска и отладки системы, так как пошаговая систематизация дает возможность определить аварийный этап и в дальнейшем нивелировать его, точно понимая, что остальные аспекты либо уже отлажены, либо не затронуты возникшей проблемой.

Первый шаг, подразумевающий подачу напряжения с требуемого блока питания, достаточно прост относительно последующих этапов, так как индицирующие светодиоды дают информацию о готовности к работе.

Второй шаг, является самым коротким относительно остальных, однако требует заострения внимания, так как при работе с различными ПЛК требуется понимать токовую развязку для входных и выходных вводов. Некоторые модели контроллеров Siemens имеют внутреннюю развязку питания для входных и выходных линий, примером такого контроллера может служить Siemens LOGO! V6 (рисунок 28).



Рисунок 28 – Siemens LOGO! V6

На данном ПЛК не вооруженным глазом заметно отсутствие выводов для подачи питания «М» и «L+». На данную особенность при проектировании стоит обращать внимание в первую очередь. Иванов В.Э пишет, что «L1 обычно обозначает вывод источника питания, не связанный с корпусом GND. N-провод подключен между полюсом L1 и цифровой клеммой. Никаких мероприятий по согласованию уровней напряжения или устранению дребезга нет, поскольку согласующее устройство уже включено в состав LOGO! [14, с. 12]»

Третьим шагом является проверка распиновки выводов блока реле. На рисунке 22 можно заметить способ механически-верного подключения без повреждения гнезда рассчитанного для зажатия одного обжатого провода, способ заключается в возможности обжатия двух проводов один без потери площади контакта и увеличения толщины кабелей, подключаемых в гнездо.

Проверка коммутации входных выводов и подключения кабеля Ethernet – четвертый и пятый шаги, также выполняются при помощи индуцирующих встроенных светодиодов в ПЛК.

Шаги пуско-наладочного процесса номер 6,7,8 следует начинать только после синхронизации TiaPortal Step7 с контроллером посредством кабеля Ethernet.

2.1.2 Программный процесс

После создания нового проекта (рисунок 29) путем выбора вкладки «Create new project», ввода названия проекта и выбора версии в которой будет сохранен проект, требуется выбрать модель контроллера, с которым будет происходить работа.

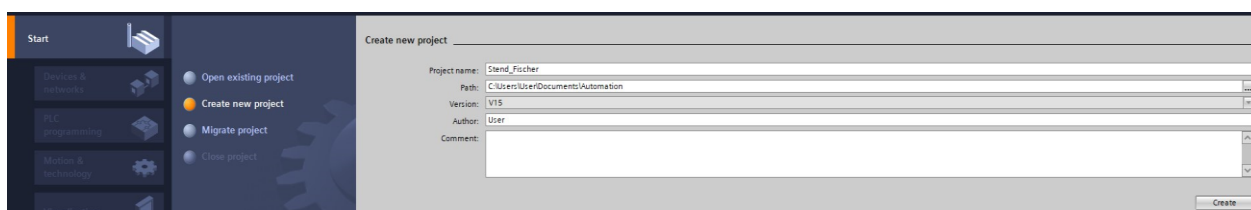


Рисунок 29 – Вкладка нового проекта в TiaPortal

Выбор контроллера из представленных моделей происходит для того чтобы среда программирования Step7 имела аппаратную отладку не только под маску используемого IP контроллера, но и под возможности аппаратной настройки входных и выходных выводов, а также их количества. Пример выбора представлен на рисунке 30. Также данная вкладка дает возможность выбрать блок дополнительных входных или выходных выводов, которые вместе конфигурируются в программе на программно-аппаратной рейке (рисунок 31) составных элементов управления, но для данного стенда достаточно встроенных выводов. «Станция S7-1200 состоит из одной стойки с максимальным количеством слотов 32. Его можно разделить на целых три “сегмента мощности”.[24, с. 48] »

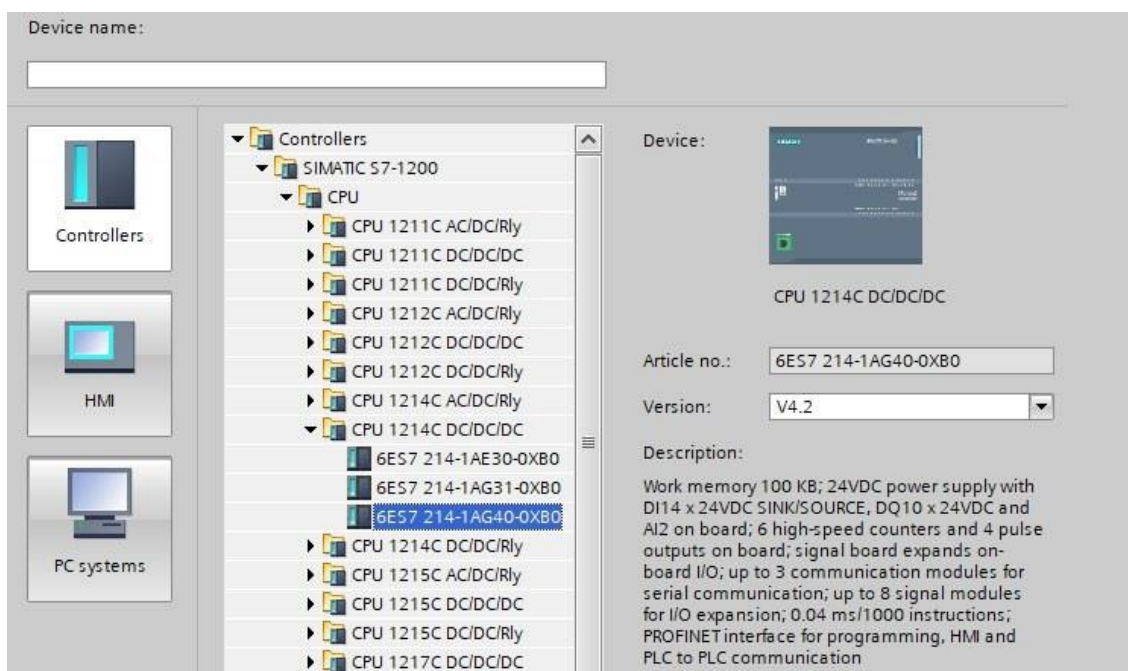


Рисунок 30 – Пример выбора программной конфигурации контроллера при создании нового проекта

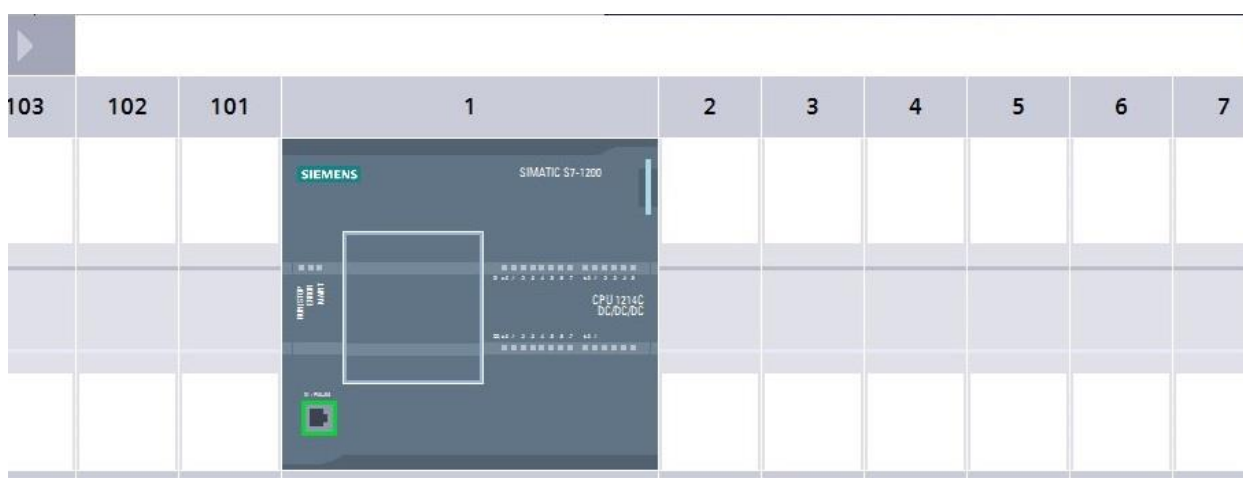


Рисунок 31 – Программно-аппаратная рейке конфигурации проекта

Справа располагаются, в случае требования конфигурацией, расширительные блоки. Слева располагается множество контроллеров, так как индекс единица присвоен ПЛК, а следующее далее цифры означают количественный номер контроллера.

Следующим шагом является сопряжения персонального компьютера с контроллером посредством Tiaportal и кабеля Ethernet. Для этого требуется нажать на вкладку «Go online» и далее откроется окно выбора подключения (рисунок 32).

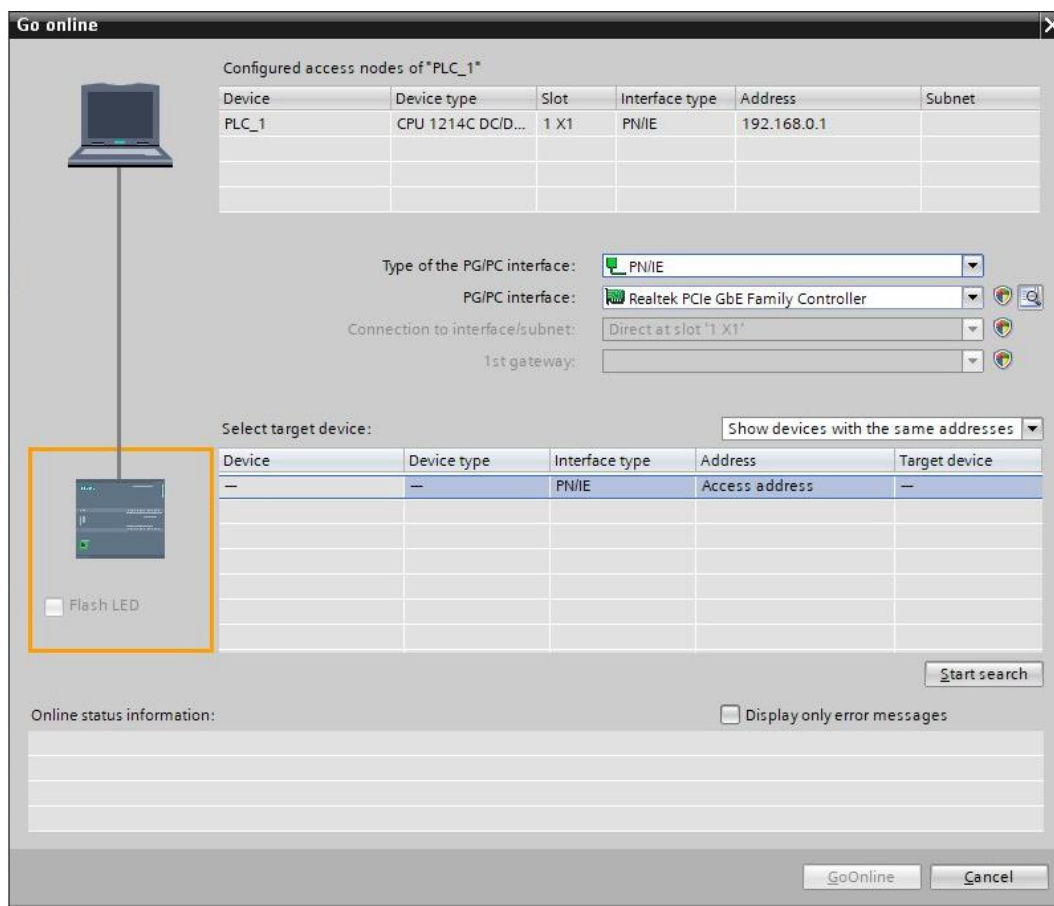


Рисунок 32 – Вкладка выбора параметров сопряжения

В типе подключения следует выбрать PN/IE так как способ подключения по беспроводной сети или телефонной линии в способе подключения через кабель Ethernet не подходит. Также следует верно выбрать сетевую карту персонального компьютера. Важнейшей точкой наладки является IP адрес контроллера, который указан в его документации и требуется для сопряжения, однако можно включить автоматизированный поиск, который автоматически считывает IP и его не будет требоваться вбивать вручную.

Заключительным этапом сопряжения является выбор в списке найденных и подключенных контроллеров требуемого для сопряжения и программирования. (рисунок 33)

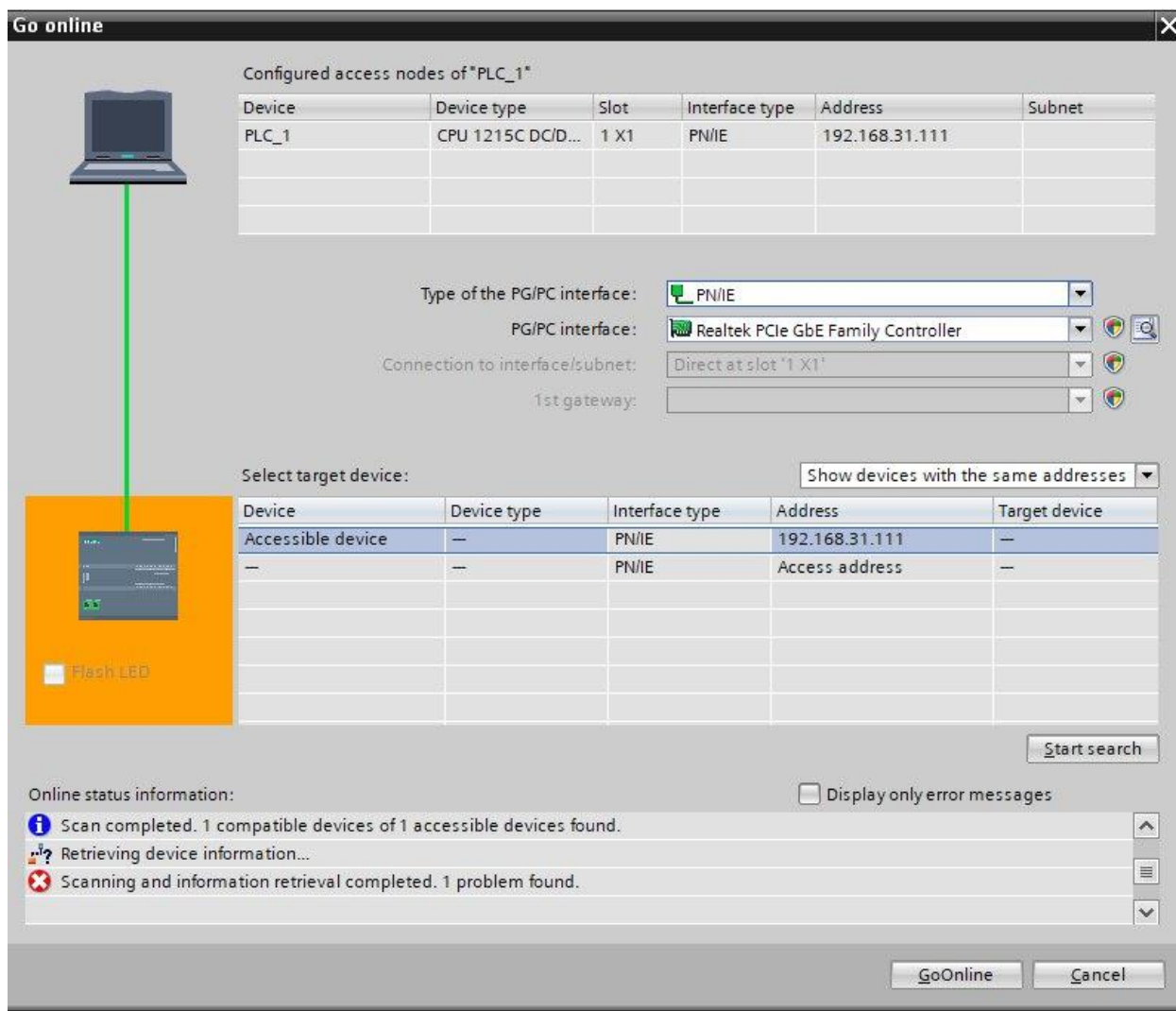


Рисунок 33 – Заклучительный этап сопряжения контроллера

После подключения ПЛК, с него можно выгрузить аппаратную и программную конфигурацию в проект для редактирования. Стоит отметить что прошивка аппаратной составляющей, которая включает в себя глубокие аппаратные назначения, такие как объявления функций скоростного счетчика в случае данного стенда, должна проводиться при подключенном к ПК контроллеру, но отключенном от исполнения процессов.

2.2 Программа автоматизированного управления стандом

Для того чтобы приступить к написанию программы на языке LAD требуется алгоритм общей системы работы и алгоритмы подсистем счета и коммутации.

2.2.1 Алгоритм работы

Алгоритм работы станда следующей

1) Нажимается кнопка работы станда, именуемая «Green_start», и активируется начало всех алгоритмов подсистем.

2) Проводится проверка замыкания концевиков нулевого положения, если ответ неудовлетворительный, то срабатывают реле обратного включения для движения в нулевое положение. Когда условие выполнено системы выходит из цикла проверки нулевого положения и начинает работу.

3) Выбирается место на «складе» от 1 до 6 соответствующей кнопкой расположения на блоке управления.

4) Проводится проверка выбранного места, как было описано ранее на «складе» имеется две координаты по горизонтальной оси и три координаты по вертикальной оси, соответственно места расположения будут работать с заданной координатой пока не достигнут ее, сами значения координат хранятся в ПЗУ контроллера, а ячейка памяти, отводимая на переменную счета для сравнения с константой координаты, является «оперативной», так как обнуляется при включение контроллера и коммутации концевиков нулевого положения.

5) Запускается цикл в котором коммутируется реле и к переменной счета одного из энкодеров прибавляется количество импульсов от энкодера, которое сравнивается с константой выбранной координаты. Пока число не достигнуто реле коммутируется, электродвигатель двигает систему.

6) По достижению выбранной координаты срабатывает цикл на выдвижение, подъем тела и возвращение в положение движения ковша.

7) Ожидание системой выбора координаты для отдачи фишки.

8) Выбор местоположения на «складе» пользователем.

9) Сравнение переменных счета по горизонтальной и вертикальной оси с выбранной координатой местоположения отгрузки фишки. Если переменная счета больше, то срабатывает реле обратного включения, запускается цикл сравнения переменной счета с константой расположения и последующим вычитанием из переменной в теле цикла, если переменная счета больше, то срабатывает реле прямого включения, запускается цикл сравнения переменной счета с константой расположения и последующим прибавлением в переменную в теле цикла.

10) Остановка по достижении выбранной координаты, срабатывание цикла на выдвижение, опускание тела и возвращение в положение движения ковша.

11) Возвращение в нулевое положение или выбор новой координаты для совершения цикла продвижения к ней, забора фишки, выбора нового местоположения, движение до новой координаты, сброс фишки.

Основной алгоритм включает в себя переменные из подпрограмм, выполняемых параллельно. Основной алгоритм представлен на рисунке 34.

«Структура проекта для системы Siemens определяется особенностями технологического процесса, который нужно автоматизировать, и конфигурацией имеющегося оборудования. Прежде всего, необходимо разбить автоматизируемый процесс на ряд простых подзадач. Для каждой частной задачи формируются требования к аппаратному и программному обеспечению [3, с. 47]»

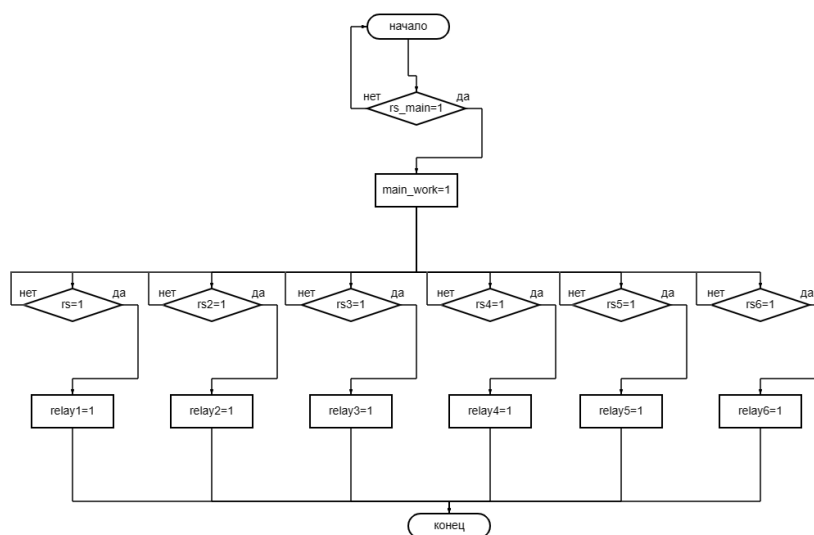


Рисунок 34 – Основной алгоритм работы стенда

Вся система стенда построена на RS триггерах, так как данный триггер отличается своей простотой в таблице истинности, что дает понятное пользователю в использование систему коммутации сигналов на управляющие стенд реле. «Функция памяти установлена (или, точнее, установлен бинарный операнд над блочным элементом памяти), если вход установки имеет сигнальное состояние «1», а вход сброса имеет сигнальное состояние «0». Функция памяти сброшена, если «1» находится на входе сброса, а «0» - на входе установки. Сигнальное состояние «0» на обоих входах не влияет на функцию для работы с памятью. Если оба входа одновременно равны «1», то две имеющиеся функции памяти реагируют по-разному: функция памяти SR сбрасывается, а функция памяти RS – устанавливается. [5, с. 203]»

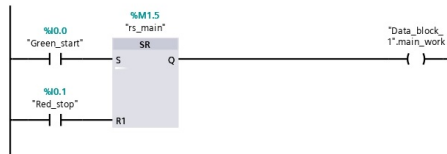
В данной программе переменные, имеющие название «rs...», являются переменными отводимыми системой памяти под значение выхода данного триггера. Переменная «main_work» является фундаментальной в данной программе, так как на нее опираются все подсистемы счета и движения для всех триггеров. В случае, когда переменная «main_work» не равняется истине, а это достигается при нажатии красной кнопки остановки на блоке клавишного управления, система останавливается и останавливаются все процессы движения.

Переменные «relay...» это соответственно булевы выходные переменные для управляющих реле, если переменная равняется истине, то реле коммутируется.

Основной алгоритм с рисунка 34 представленный на языке LAD в системе программирования TiaPortal v15 представлен на рисунке 35.

«Редактор LAD определяет сегмент в соответствии с принципом «главного звена». Это самая верхняя ветвь, которая начинается непосредственно на левой несущей (питающей шине) и должна завершаться катушкой или блочным элементом. В этом звене могут быть расположены все элементы LAD. В параллельных ветвях, которые не берут начало на левой несущей (питающей шине), иногда действуют ограничения в зависимости от тех или иных программных элементов. Дополнительные ограничения определяют следующее: элемент LAD не может быть «замкнут накоротко» с «пустой» параллельной ветвью, и «ток» не может протекать через элемент справа налево (параллельная ветвь должна быть замкнута на ветвь, в которой она была разомкнута) [5, с. 144]»

Network 1:



Network 2:

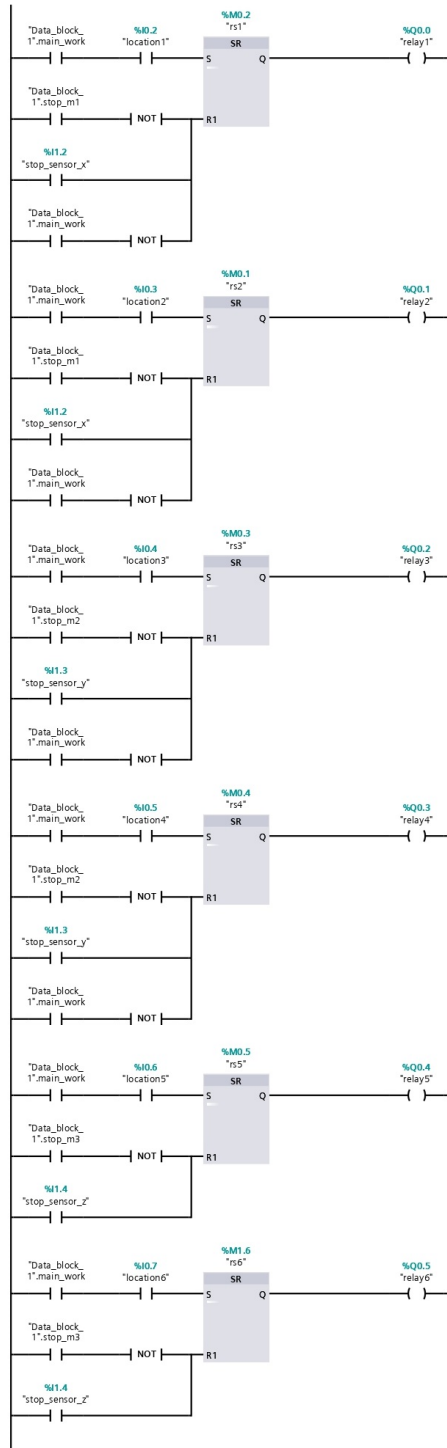


Рисунок 35 – Основной алгоритм работы системы станда на языке LAD

Основной алгоритм обеспечивает работу общей силовой части, которая управляется ПЛК, однако основная часть расчетов по управлению проходит в алгоритмах и программах подсистем осей горизонтальной, вертикальной и выдвигной.

Развитие программы начинается с фундаментального передвижения, которое далее модернизируется в цикл запросов координаты, расчета и передвижения к ней. Так базовая модель алгоритма движения в нулевой момент времени выглядит следующим образом (рисунок 36).

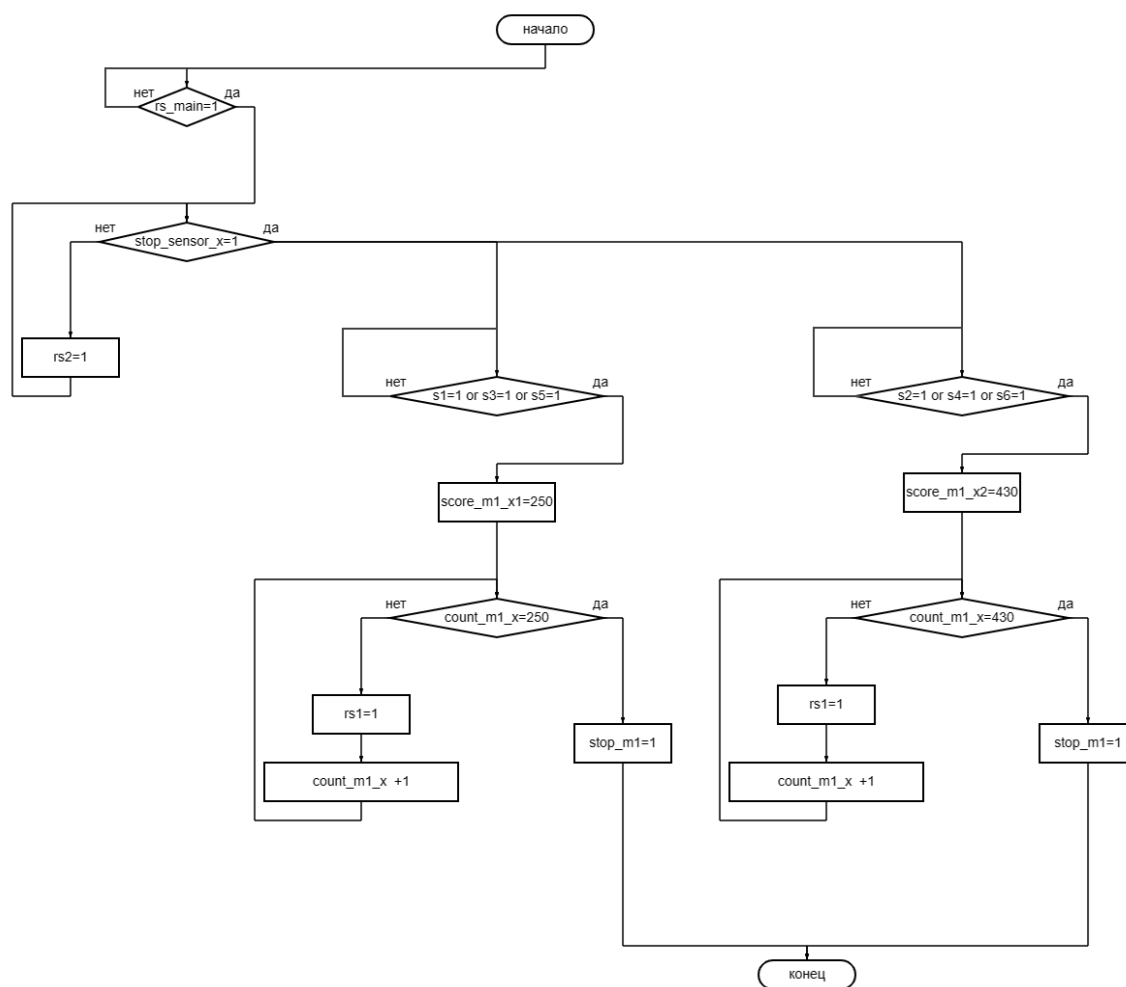


Рисунок 36 – Базовая модель алгоритма движения по горизонтальной оси

Сначала проходит проверка переменной нулевого положения, если она не равна истине, то включаете реле обратного хода до тех пор, пока контакт не

замкнется и переменной не присвоится единица. Далее начинается алгоритм сравнения константы координаты с переменной счета, в зависимости от выбранного столбца как на рисунке 3. Но данный алгоритм завершается по достижению выбранной координаты для того чтобы алгоритм работал до тех пор, пока выбирается координата следует включить в алгоритм такую переменную как ожидание и цикл, который не будет прекращаться пока пользователь не нажмет на кнопки остановки для переменной «rs_main» (main_work = 0). Модернизированный алгоритм представлен на рисунке 37.

В обновленный алгоритм помимо блока ожидания, были добавлены переменные «s_n» и «s_out». Переменная «s_n» присваивает себе значение одной из двух координат по которой двигался алгоритм, в дальнейшем значение этой переменной переписывается в зависимости от занятой координаты после завершения тела цикла, эта переменная нужна для понимания алгоритма в какую сторону вести счет, поэтому также введена переменная «s_out», которая является выбираемой переменной, которая сравнивается с «s_n» и выбирает линию прибавления значения в ячейку счета если «s_out» больше «s_n» или линию вычитания координаты в случае если «s_out» меньше «s_n». Чтобы алгоритм имел выход из цикла,0 при возвращении в точку ожидания введена проверка главного выключателя переменная которого именуется «main_work». Стоит отметить что в программе LAD проверка всех точек разрешения происходит параллельно друг от друга, и завершение процесса работы может произойти не только по достижению координаты, но и в любой точке алгоритма.

«LAD или Ladder logic - это основной метод программирования, используемый для ПЛК. Как упоминалось ранее, лестничная логика была разработана для имитации логики ретрансляции [25, с. 21]»

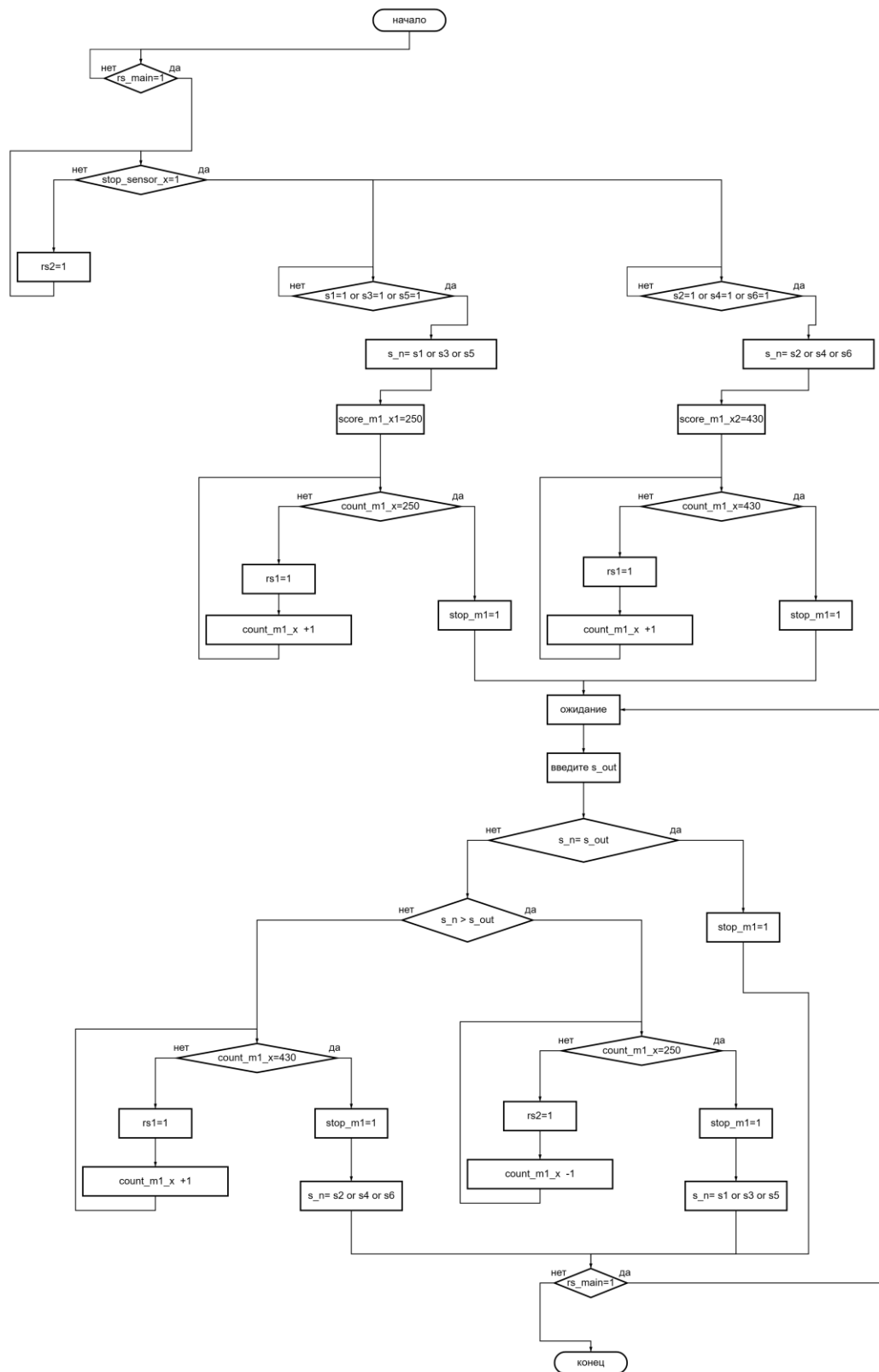


Рисунок 37 – Модернизированный алгоритм движения с циклом

Алгоритм для вертикальной оси усложнен относительно алгоритма для горизонтальной оси по той причине, что в работе по вертикали присутствуют

три координаты, а также подсистема подъема и отдачи фишки которая должна срабатывать в тот момент, когда манипуляционный хват достиг требуемых координат, также следует проводить проверку переменной захвата во избежание двойного подряд захвата или двойного сброса фишки, вместо поочередной работы данных функций. Так как алгоритм для вертикальной оси достаточно большой он разбит на две части, в первой части представлен алгоритм передвижения во время включения системы (рисунок 38), во второй части представлен основной цикл работы, который работает по выше описанному принципу сравнения, но с добавкой конкретизации насколько отличается координата нынешнего положения и последующего, а также с проверкой в какую - в большую или меньшую сторону находится отклонение (рисунок 39).

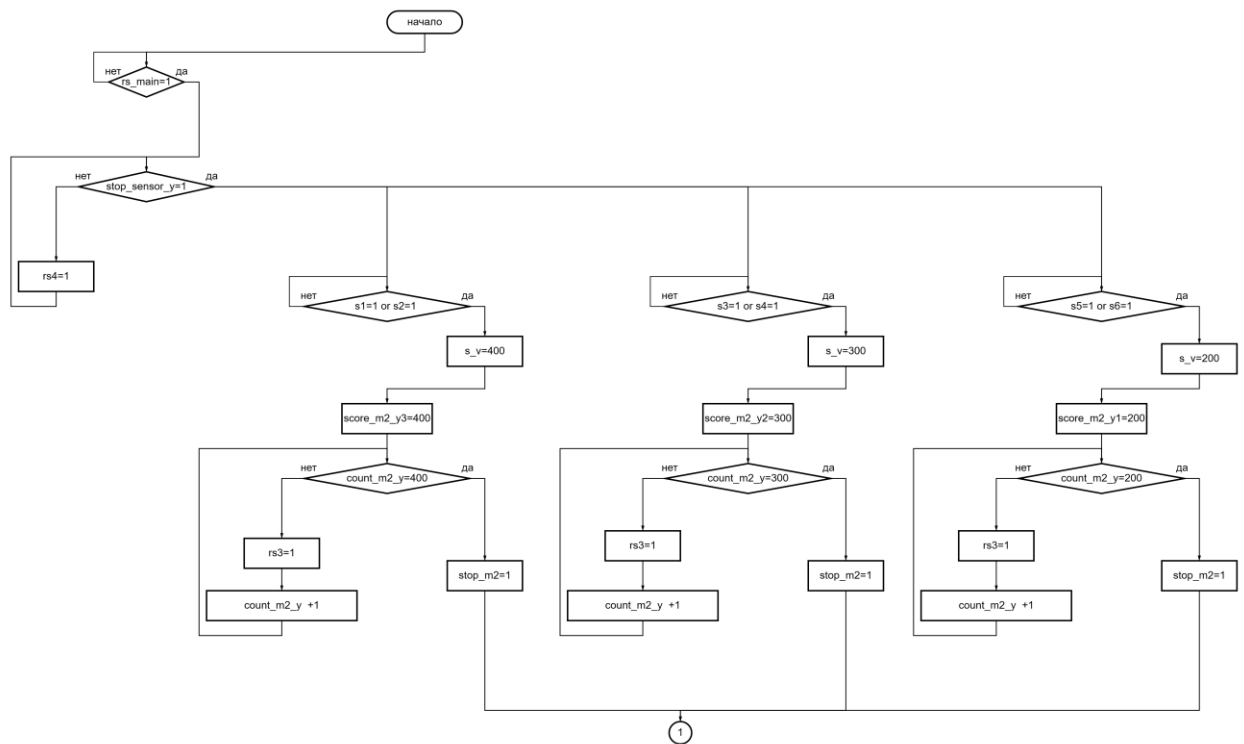


Рисунок 38 – Часть алгоритма для вертикальной оси отвечающей за начало движения после включения

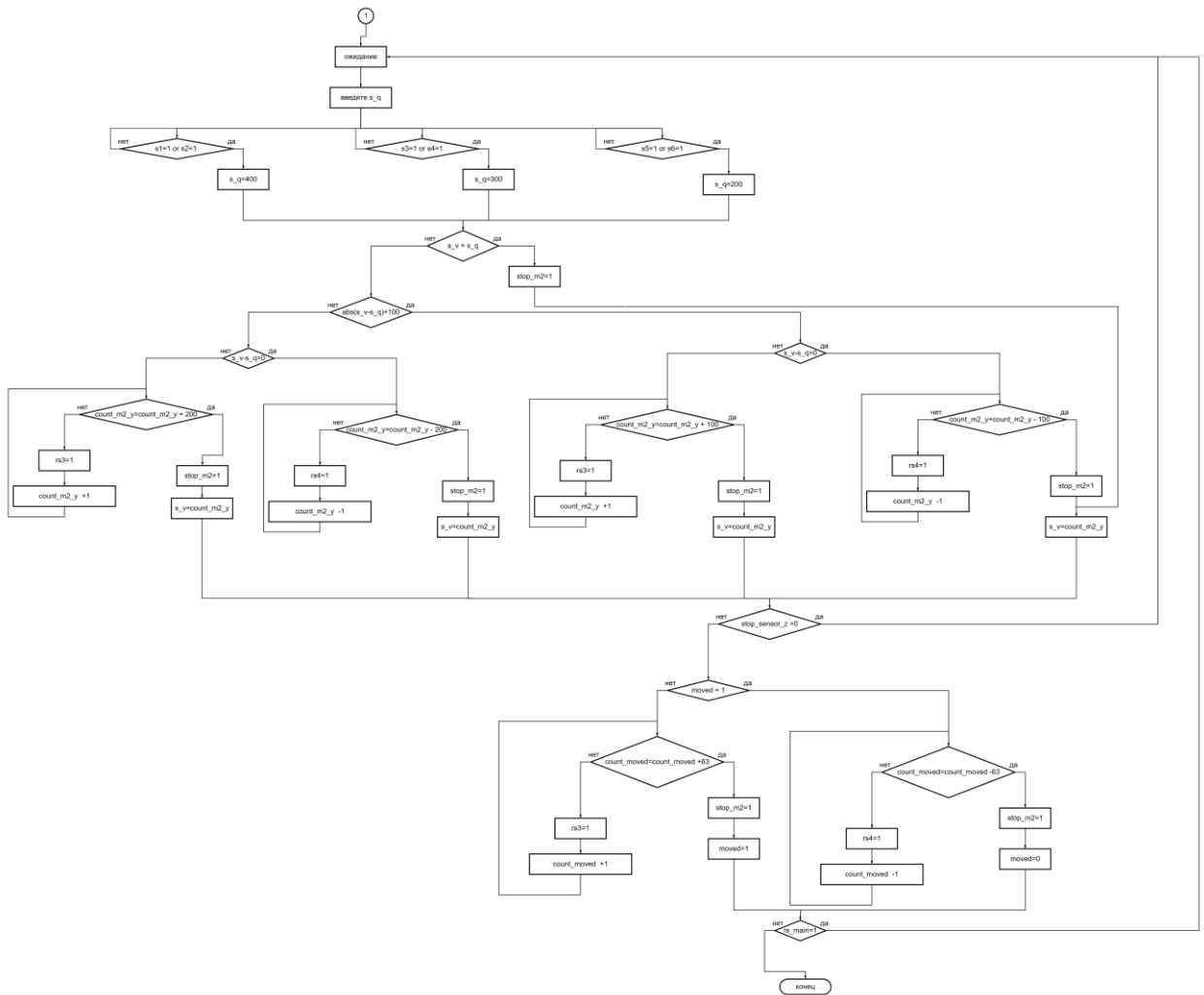


Рисунок 39 – Основной цикл движения по «складу» с подъемом и спуском фишки

Цикл подъема и спуска фишки разделен с циклом движения по «складу» проверкой конечного выключателя, который отвечает за выдвигание хвата манипулятора. Иначе говоря, работа со спуском или поднятием фишки не начнется если хват не выдвинулся и не замкнул кнопку полного выдвигания. Данная переменная «stop_sensor_z» является переменной алгоритма для оси выдвигания (рисунок 40), и связывает эти два алгоритма между собой.

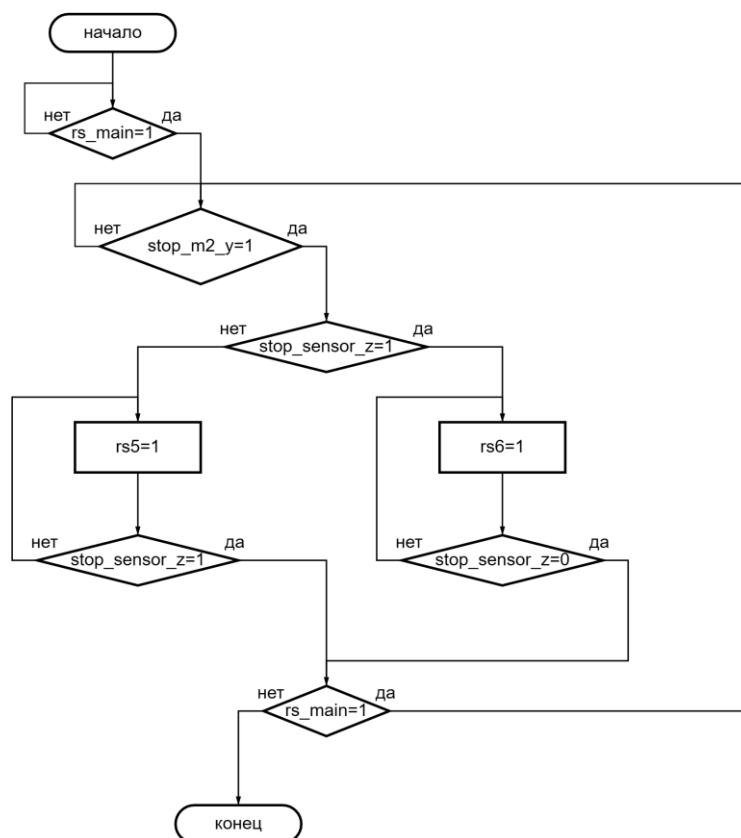


Рисунок 40 – Алгоритм для оси выдвигения

Данный алгоритм проще по своей структуре, потому как его обратная связь реализована на концевых выключателях начального положения «start_sensor_z» и конечного «stop_sensor_z». Также в его структуре используется связь с подсистемой движения по вертикали «stop_m2_y», чтобы движение манипулятора не происходило при подъеме или опускании хвата, а только при его остановке.

2.2.2 LAD и SCL программы

Для того чтобы составить программу соответственно алгоритму первоначально следует ввести блок данных. Блок данных переменных делится на два типа – булевые и целочисленные. Булевой блок данных представлен на рисунке 41, а целочисленный на рисунке 42.

Data_bool							
	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...
1	Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	main_work	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	stop_m1	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	stop_m2	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	stop_m3	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	rs 1	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	rs 2	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8	rs 3	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9	rs 4	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
10	rs 5	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
11	rs 6	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
12	stop_sensor_z	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
13	moved	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Рисунок 41 – Булевой блок переменных

В блок данных переменных булевого вида входят переменные состояний необходимые для проверок и коммутаций.

Data_Int							
	Name	Data type	Start value	Retain	Accessible f...	Writable f..	Visible in ...
1	Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	count_m1_x	Int	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	count_m2_y	Int	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	score_m1_x1	Int	250	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	score_m2_y1	Int	200	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	score_m1_x2	Int	430	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	score_m2_y2	Int	300	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8	score_m2_y3	Int	400	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9	s_v	Int	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
10	s_q	Int	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
11	s1	Int	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
12	s2	Int	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
13	s3	Int	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
14	s4	Int	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
15	s5	Int	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
16	count_moved	Int	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
17	s_n	Int	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
18	s_out	Int	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Рисунок 42 – Целочисленный блок переменных

В блок данных целочисленных переменных входят переменные, которые подвергаются программным расчетам и сравнениям. Здесь также вводятся константы положений для горизонтальной и вертикальной оси.

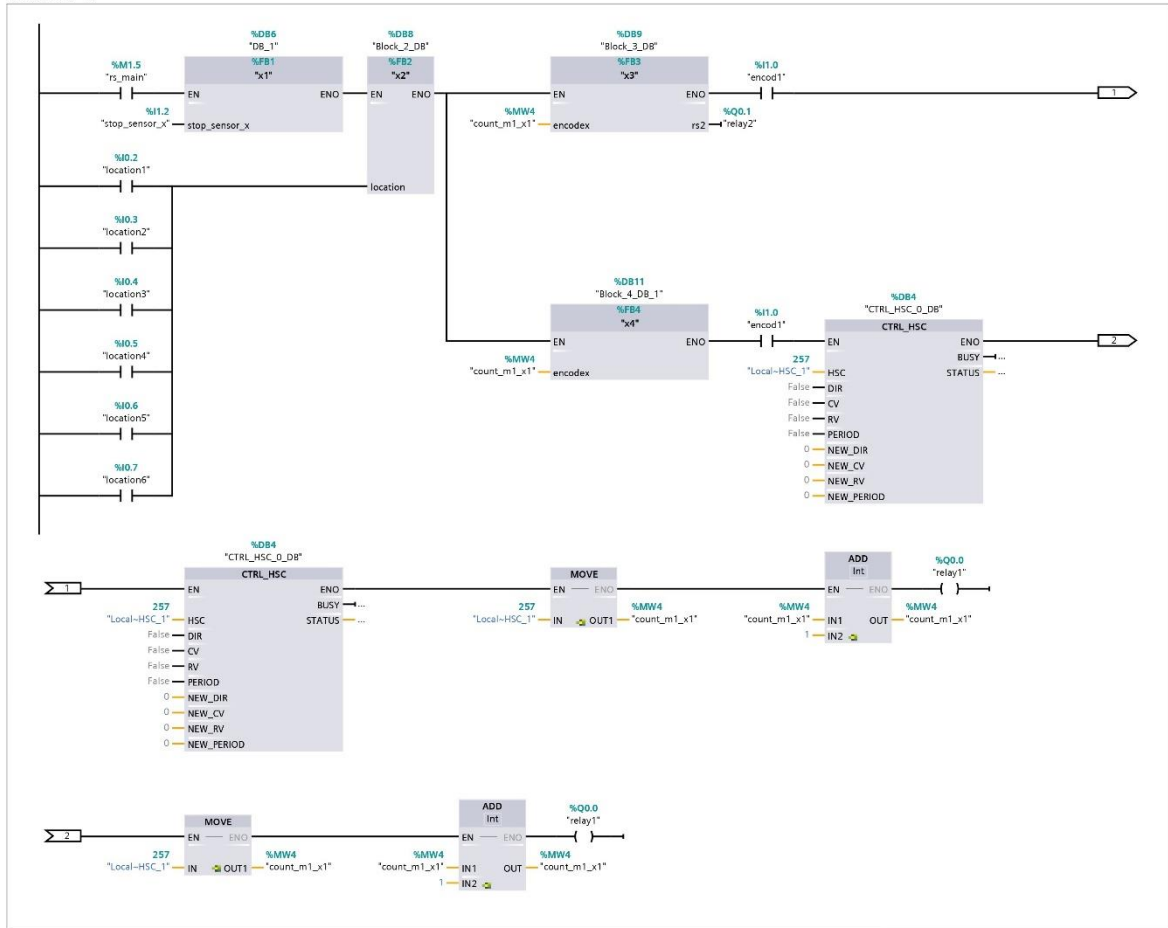
Для алгоритма для вертикальной оси с рисунка 37, составлена программа битовой логики на языке LAD (рисунок 44) с интегрированными программными блоками на языке SCL, для которых также описывается отдельный блок данных (рисунок 43), потому как битовая логика исключает введение циклов с двумя линиями ответов, которые включает в себя объемные тела циклов.

Tag table_1							
	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...
1	s_n	Int	%MW1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	stop_m1	Bool	%M4.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	count_m1_x1	Int	%MW4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	s_out	Int	%MW6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	s_v	Int	%MW8	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	count_m2_y1	Int	%MW10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	s_q	Int	%MW12	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8	stop_m2	Bool	%M14.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9	moved	Bool	%M14.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
10	count_moved	Int	%MW14	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Рисунок 43 – Таблица тэгов для подсистем

На данном рисунке 43 представлена таблица тэгов для проведения внутренних расчетов и сравнений. Данные переменные имеют адрес, начинающийся с «М», что говорит о том, что данные переменные не привязаны к входам или выходам контроллера, а будут храниться в памяти контроллера и вызываться при вызове требуемой функции в которой они содержатся. Сергеев пишет об этом следующее «Блок данных не нужен функциям, так как они находятся ниже программного блока по иерархии. Однако, параметры функций также перечисляются в окне объявления переменных, и статические данные могут быть использованы внутри функций.[19, с. 29]»

Network 1:



Network 2:

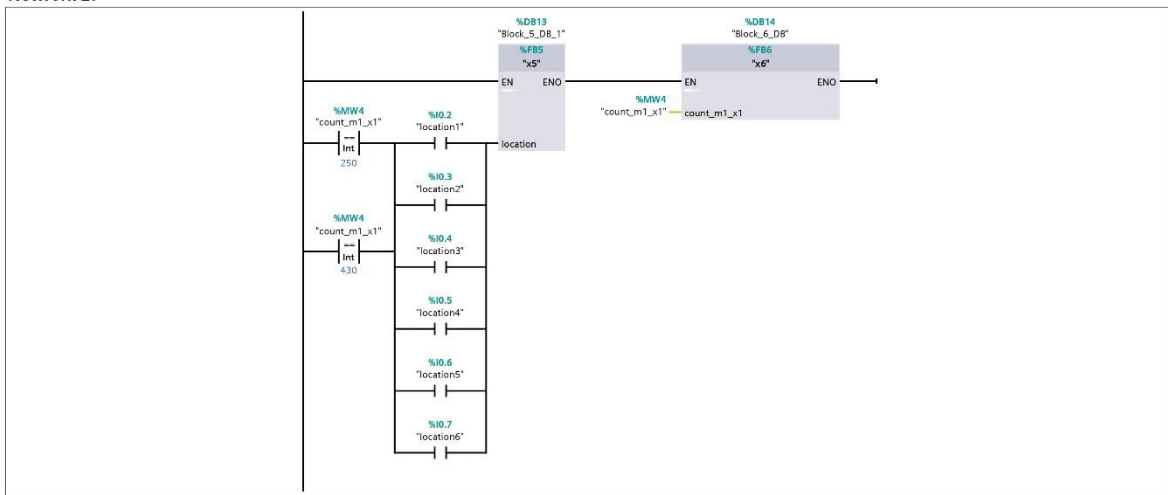


Рисунок 44 –Программа движения по горизонтальной оси на языке LAD

Программные блоки «x1-x6» описываются отдельными программами SCL, которые будут приведены ниже на рисунках 45-50.

x1 [FB1]										
x1 Properties										
General										
Name	x1	Number	1	Type	FB	Language	SCL			
Numbering	Automatic									
Information										
Title				Author				Comment	Family	
Version	0.1			User-defined ID						
Name	Data type	Default value	Retain	Accessible from HMI/OPC UA	Write-able from HMI/OPC UA	Visible in HMI engineering	Setpoint	Supervision	Comment	
▼ Input										
stop_sensor_x	Bool	false	Non-retain	True	True	True	False			
Output										
InOut										
Static										
Temp										
Constant										
<pre> 0001 0002 IF "stop_sensor_x" = 0 THEN 0003 "rs2" = 1 0004 ELSE "rs2" = 0 0005 END_IF; 0006 </pre>										
Symbol	Address	Type	Comment							
"rs2"	%M0.1	Bool								
"stop_sensor_x"	%I1.2	Bool								

Рисунок 45 – SCL код для программного блока «x1»

x2 [FB2]										
x2 Properties										
General										
Name	x2	Number	2	Type	FB	Language	SCL			
Numbering	Automatic									
Information										
Title				Author				Comment	Family	
Version	0.1			User-defined ID						
Name	Data type	Default value	Retain	Accessible from HMI/OPC UA	Write-able from HMI/OPC UA	Visible in HMI engineering	Setpoint	Supervision	Comment	
▼ Input										
location	Bool	false	Non-retain	True	True	True	False			
Output										
InOut										
Static										
▼ Temp										
s_n	Int									
▼ Constant										
score_m1_x1	Int	250								
<pre> 0001 IF "location2" OR "location4" OR "location6" = 1 THEN 0002 "s_n" = 430; 0003 END_IF; 0004 IF "location1" OR "location3" OR "location5" = 1 THEN 0005 "s_n" = 250 0006 ; 0007 END_IF; 0008 </pre>										
Symbol	Address	Type	Comment							
"location1"	%I0.2	Bool								
"location2"	%I0.3	Bool								
"location3"	%I0.4	Bool								
"location4"	%I0.5	Bool								
"location5"	%I0.6	Bool								
"location6"	%I0.7	Bool								
"s_n"	%MW1	Int								

Рисунок 46 – SCL код для программного блока «x2»

x3 [FB3]

x3 Properties									
General									
Name	x3	Number	3	Type	FB	Language	SCL		
Numbering	Automatic								
Information									
Title	Author			Comment			Family		
Version	0.1			User-defined ID					
Name	Data type	Default value	Retain	Accessible from HMI/OPC UA	Write-able from HMI/OPC UA	Visible in HMI engineering	Setpoint	Supervision	Comment
▼ Input									
encodex	Int	0	Non-retain	True	True	True	False		
▼ Output									
rs2	Bool	false	Non-retain	True	True	True	False		
InOut									
Static									
▼ Temp									
count_m1_x1	Int								
Constant									
<pre> 0001 IF "count_m1_x1" = 250 THEN 0002 "stop_m1" = 1 0003 ELSE "rs2" = 1 0004 END_IF; 0005 </pre>									
Symbol	Address	Type	Comment						
"count_m1_x1"	%MW4	Int							
"rs2"	%M0.1	Bool							
"stop_m1"	%M4.0	Bool							

Рисунок 47 – SCL код для программного блока «x3»

x4 [FB4]

x4 Properties									
General									
Name	x4	Number	4	Type	FB	Language	SCL		
Numbering	Automatic								
Information									
Title	Author			Comment			Family		
Version	0.1			User-defined ID					
Name	Data type	Default value	Retain	Accessible from HMI/OPC UA	Write-able from HMI/OPC UA	Visible in HMI engineering	Setpoint	Supervision	Comment
▼ Input									
encodex	Int	0	Non-retain	True	True	True	False		
Output									
InOut									
Static									
Temp									
Constant									
<pre> 0001 IF "count_m1_x1" = 430 THEN 0002 "stop_m1" = 1 0003 ELSE "rs2" = 1 0004 END_IF; 0005 </pre>									
Symbol	Address	Type	Comment						
"count_m1_x1"	%MW4	Int							
"rs2"	%M0.1	Bool							
"stop_m1"	%M4.0	Bool							

Рисунок 48 – SCL код для программного блока «x4»

x5 [FB5]

x5 Properties									
General									
Name	x5	Number	5	Type	FB	Language	SCL		
Numbering	Automatic								
Information									
Title	Author			Comment			Family		
Version	0.1	User-defined ID							
Name	Data type	Default value	Retain	Accessible from HMI/OPC UA	Write-Visible in HMI/OPC UA	Visible in engineering	Setpoint	Supervision	Comment
▼ Input									
location	Bool	false	Non-retain	True	True	True	False		
Output									
InOut									
Static									
Temp									
Constant									
<pre> 0001 IF "location2" OR "location4" OR "location6" = 1 THEN 0002 "s_out" = 430 0003 ; 0004 END_IF; 0005 IF "location1" OR "location3" OR "location5" = 1 THEN 0006 "s_out" = 250 0007 ; 0008 END_IF; 0009 </pre>									
Symbol	Address	Type	Comment						
"location1"	%I0.2	Bool							
"location2"	%I0.3	Bool							
"location3"	%I0.4	Bool							
"location4"	%I0.5	Bool							
"location5"	%I0.6	Bool							
"location6"	%I0.7	Bool							
"s_out"	%MW6	Int							

Рисунок 49 – SCL код для программного блока «x5»

x6 [FB6]

x6 Properties									
General									
Name	x6	Number	6	Type	FB	Language	SCL		
Numbering	Automatic								
Information									
Title	Author			Comment			Family		
Version	0.1	User-defined ID							
Name	Data type	Default value	Retain	Accessible from HMI/OPC UA	Write-Visible in HMI/OPC UA	Visible in engineering	Setpoint	Supervision	Comment
▼ Input									
count_m1_x1	Int	0	Non-retain	True	True	True	False		
Output									
InOut									
Static									
Temp									
Constant									
<pre> 0001 IF "s_n" = "s_out" THEN 0002 "stop_m1" = 1 0003 ; 0004 ELSE IF "s_n" > "s_out" THEN 0005 IF "count_m1_x1" = 250 THEN 0006 "stop_m1"=1; 0007 "s_n" = 250; 0008 ELSE "rs2"=1; "count_m1_x1" -1; 0009 END_IF 0010 IF IF "count_m1_x1" = 430 THEN 0011 "stop_m1"=1; 0012 "s_n" = 430; 0013 ELSE "rs2"=1; "count_m1_x1" +1; 0014 END_IF 0015 END_IF; 0016 </pre>									
Symbol	Address	Type	Comment						
"count_m1_x1"	%MW4	Int							
"rs2"	%M0.1	Bool							
"s_n"	%MW1	Int							
"s_out"	%MW6	Int							
"stop_m1"	%M4.0	Bool							

Рисунок 50 – SCL код для программного блока «x6»

Все переменные используемые в строковой программе по своему названию совпадают с переменными из алгоритма для удобства ориентирования по программе.

Программа LAD для вертикальной оси обширнее по сравнению с программой для вертикальной оси, поэтому она также разбита на две подсети «Network 1» (рисунок 51) и «Network 2» (рисунок 52), но вторая подсеть приведена отдельным рисунком.

Так как работа данной системы программирования подразумевает работу всех подсетей одновременно, для того чтобы вторая подсеть, которая запрашивает вторичное положение после приезда на начальную точку для выполнения цикла с поднятием и спуском фишки, запрашивала положение для второй локации для снятия или подъема фишки, вторая подсеть с циклом была отделена от общей системы условием эквивалентности переменной расчета «count_m2_y» одной из трех констант координаты для первого, второго и третьего уровня соответственно.

В подсистеме расчета и движения по вертикальной оси введено семь программных блоков SCL «y1-y7» (рисунок 53-59), которые включают в себя подобный горизонтальной оси код, за исключением усложненного цикла, который также опирается на переменные из оси выдвигания манипулятора и имеет более усложненный расчет сравнения предыдущей координаты и выбираемой оператором для промежуточно-конечной точки.

«Счетчик выполняет прямой счет ("инкрементируется"), если инструкция CU обрабатывается при положительном (возрастающем) фронте сигнала RLO (RLO меняет свое состояние с "0" на "1"). Каждый положительный фронт сигнала, предшествующий операции CU, увеличивает значение счетчика ("count value") на единицу, пока не будет достигнут верхний

предел, равный 999. После этого положительный фронт сигнала RLO перед вводом CU никак не будет влиять на состояние счетчика. [4, с. 260]»

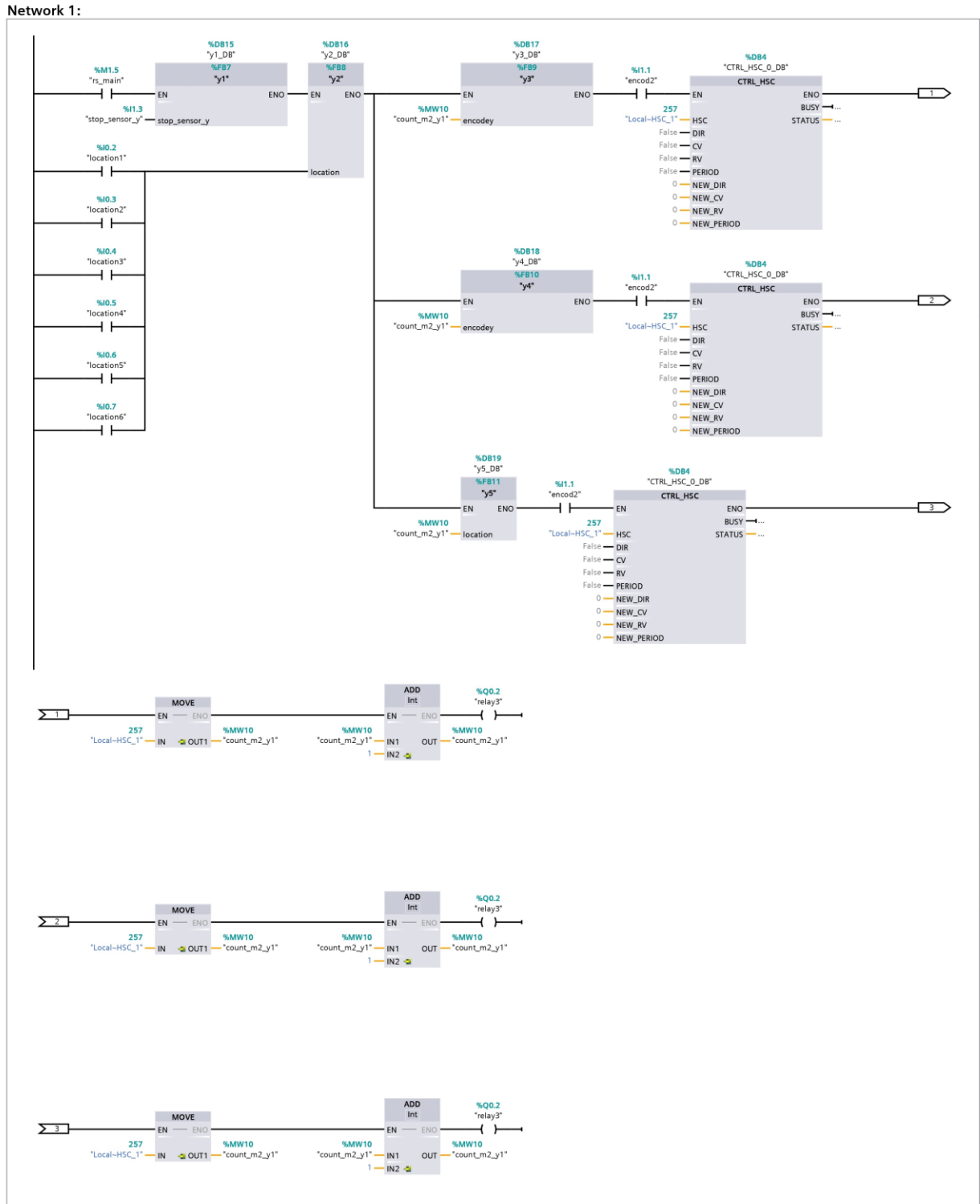


Рисунок 51 – Программа движения по вертикальной оси на языке LAD первой подсети

Network 2:

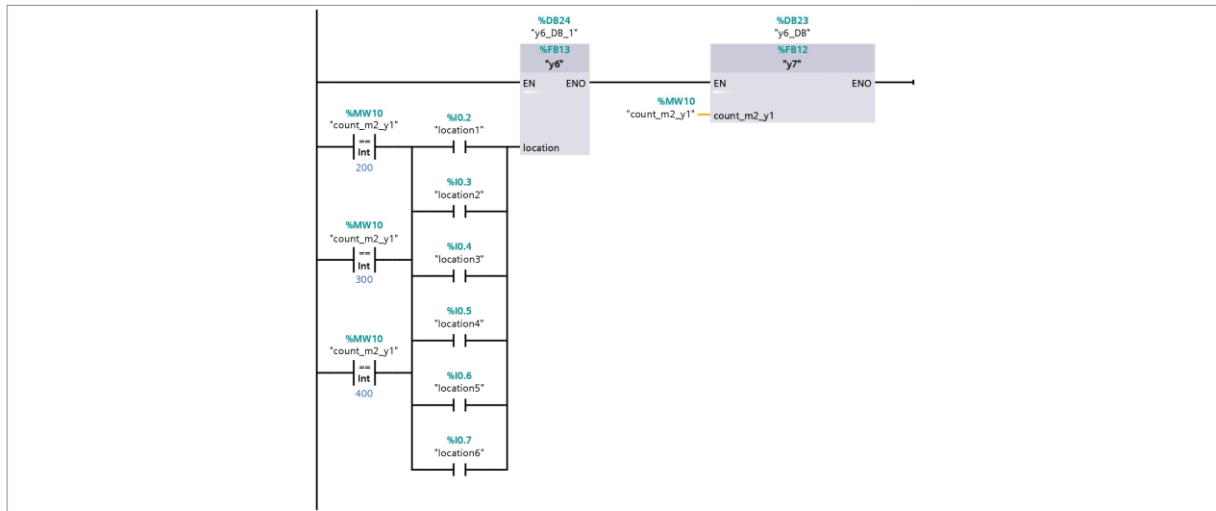


Рисунок 52 – Программа движения по вертикальной оси на языке LAD второй подсети

y1 [FB7]

y1 Properties									
General									
Name	y1	Number	7	Type	FB	Language	SCL		
Numbering	Automatic								
Information									
Title		Author		Comment		Family			
Version	0.1	User-defined ID							
Name	Data type	Default value	Retain	Accessible from HMI/OPC UA	Writ-Visible in HMI/OPC UA	Setpoint	Supervision	Comment	
▼ Input									
stop_sensor_y	Bool	false	Non-retain	True	True	True	False		
Output									
InOut									
Static									
Temp									
Constant									
<pre> 0001 0002 IF "stop_sensor_y" = 0 THEN 0003 "rs4" = 1 0004 ELSE "rs4" = 0 0005 END_IF; 0006 </pre>									
Symbol	Address	Type	Comment						
"rs4"	%M0.4	Bool							
"stop_sensor_y"	%I1.3	Bool							

Рисунок 53 – SCL код для программного блока «y1»

y2 [FB8]

y2 Properties									
General									
Name	y2	Number	8	Type	FB	Language	SCL		
Numbering	Automatic								
Information									
Title		Author		Comment		Family			
Version	0.1	User-defined ID							
Name	Data type	Default value	Retain	Accessible from HMI/OPC UA	Write-Visible from HMI/OPC UA	Visible in HMI engineering	Setpoint	Supervision	Comment
▼ Input									
location	Bool	false	Non-retain	True	True	True	False		
Output									
InOut									
Static									
Temp									
Constant									
<pre> 0001 IF "location1" OR "location2" = 1 THEN 0002 "s_v" = 200 0003 ; 0004 END_IF; 0005 IF "location3" OR "location4" = 1 THEN 0006 "s_v" = 300 0007 ; 0008 END_IF; 0009 IF "location5" OR "location6" = 1 THEN 0010 "s_v" = 400 0011 ; 0012 END_IF; 0013 </pre>									
Symbol	Address	Type	Comment						
"location1"	%I0.2	Bool							
"location2"	%I0.3	Bool							
"location3"	%I0.4	Bool							
"location4"	%I0.5	Bool							
"location5"	%I0.6	Bool							
"location6"	%I0.7	Bool							
"s_v"	%MW8	Int							

Рисунок 54 – SCL код для программного блока «y2»

y3 [FB9]

y3 Properties									
General									
Name	y3	Number	9	Type	FB	Language	SCL		
Numbering	Automatic								
Information									
Title		Author		Comment		Family			
Version	0.1	User-defined ID							
Name	Data type	Default value	Retain	Accessible from HMI/OPC UA	Write-Visible from HMI/OPC UA	Visible in HMI engineering	Setpoint	Supervision	Comment
▼ Input									
encodey	Int	0	Non-retain	True	True	True	False		
Output									
InOut									
Static									
Temp									
Constant									
<pre> 0001 IF "count_m2_y1" = 200 THEN 0002 "stop_m2" = 1 0003 ELSE "rs3" = 1 0004 END_IF; 0005 </pre>									
Symbol	Address	Type	Comment						
"count_m2_y1"	%MW10	Int							
"rs3"	%M0.3	Bool							
"stop_m2"	%M14.0	Bool							

Рисунок 55 – SCL код для программного блока «y3»

y4 [FB10]

y4 Properties									
General									
Name	y4	Number	10	Type	FB	Language	SCL		
Numbering	Automatic								
Information									
Title		Author		Comment		Family			
Version	0.1	User-defined ID							
Name	Data type	Default value	Retain	Accessible from HMI/OPC UA	Write-Visible from HMI/OPC UA	Visible in HMI engineering	Setpoint	Supervision	Comment
▼ Input									
encodey	Int	0	Non-retain	True	True	True	False		
Output									
InOut									
Static									
Temp									
Constant									
<pre> 0001 IF "count_m2_y1" = 300 THEN 0002 "stop_m2" = 1 0003 ELSE "rs3" = 1 0004 END_IF; 0005 </pre>									
Symbol	Address	Type	Comment						
"count_m2_y1"	%MW10	Int							
"rs3"	%M0.3	Bool							
"stop_m2"	%M14.0	Bool							

Рисунок 56 – SCL код для программного блока «y4»

y5 [FB11]

y5 Properties									
General									
Name	y5	Number	11	Type	FB	Language	SCL		
Numbering	Automatic								
Information									
Title		Author		Comment		Family			
Version	0.1	User-defined ID							
Name	Data type	Default value	Retain	Accessible from HMI/OPC UA	Write-Visible from HMI/OPC UA	Visible in HMI engineering	Setpoint	Supervision	Comment
▼ Input									
location	Int	0	Non-retain	True	True	True	False		
Output									
InOut									
Static									
Temp									
Constant									
<pre> 0001 IF "count_m2_y1" = 400 THEN 0002 "stop_m2" = 1 0003 ELSE "rs3" = 1 0004 END_IF; 0005 </pre>									
Symbol	Address	Type	Comment						
"count_m2_y1"	%MW10	Int							
"rs3"	%M0.3	Bool							
"stop_m2"	%M14.0	Bool							

Рисунок 57 – SCL код для программного блока «y5»

«Каждая операция команды или данных выполняется за один период частоты дискретизации. Все команды требуют по крайней мере еще одного периода дискретизации после первоначального чтения функции. [17, с. 301]»

y6 [FB13]

y6 Properties									
General									
Name	y6	Number	13	Type	FB	Language	SCL		
Numbering	Automatic								
Information									
Title	Author			Comment			Family		
Version	0.1	User-defined ID							
Name	Data type	Default value	Retain	Accessible from HMI/OPC UA	Write-Visible in HMI/OPC UA	Visible in engineering	Setpoint	Supervision	Comment
▼ Input									
location	Bool	false	Non-retain	True	True	True	False		
Output									
InOut									
Static									
Temp									
Constant									
<pre> 0001 IF "location1" OR "location2" = 1 THEN 0002 "s_v" = 200 0003 ; 0004 END_IF; 0005 IF "location3" OR "location4" = 1 THEN 0006 "s_v" = 300 0007 ; 0008 END_IF; 0009 IF "location5" OR "location6" = 1 THEN 0010 "s_v" = 400 0011 ; 0012 END_IF; 0013 </pre>									
Symbol	Address	Type	Comment						
"location1"	%I0.2	Bool							
"location2"	%I0.3	Bool							
"location3"	%I0.4	Bool							
"location4"	%I0.5	Bool							
"location5"	%I0.6	Bool							
"location6"	%I0.7	Bool							
"s_v"	%MWS	Int							

Рисунок 58 – SCL код для программного блока «уб»

В программном блоке «у7» используются условия «IF» в разветвлении которых также вставляются условия. После описания каждого условия со следствиями в строке прописывается «END_IF», данная команда завершает выполнение цикла.

Анализируя проделываемую работу, можно привести перевод выдержки из статьи «Роль человека в эру автоматизации» Бредли Фарела «Автоматизация - это свидетельство научно-технической революции и одно из наиболее важных ее проявлений. Поэтому, ее невозможно рассматривать отдельно от более широких изменений в техносфере, которая включает в себя наработки информатики, математики и языков программирования [23, с. 2]»

Totally Integrated Automation Portal									
y7 [FB12]									
y7 Properties									
General									
Name	y7	Number	12	Type	FB	Language	SCL		
Numbering	Automatic								
Information									
Title		Author		Comment		Family			
Version	0.1	User-defined ID							
Name	Data type	Default value	Retain	Accessible from HMI/OPC UA	Write-Visible in HMI/OPC UA	Visible in engineering	Setpoint	Supervision	Comment
▼ Input									
count_m2_y1	Int	0	Non-retain	True	True	True	False		
Output									
InOut									
Static									
Temp									
Constant									
<pre> 0001 IF "s_v" = "s_q" THEN 0002 "stop_m2" = 1 0003 ; 0004 ELSE IF (ABS("s_v"-"s_q")=100) THEN 0005 IF "s_v"-"s_q">0 THEN 0006 IF "count_m2_y1"="count_m2_y1"-100 THEN 0007 "stop_m2"=1; 0008 "s_v" := "count_m2_y1"; 0009 IF "stop_sensor_z"=0 THEN GOTO 1 0010 ELSE IF "moved"=1 THEN 0011 IF "count_moved"="count_moved"-63 THEN "stop_m2"=1; "moved"=0; 0012 ELSE "rs4"=1; "count_moved"-1; END_IF 0013 ELSE IF "count_moved"="count_moved"+63 THEN "stop_m2"=1; "moved"=1; 0014 ELSE "rs4"=1; "count_moved"+1; END_IF 0015 ELSE "rs4"=1; "count_m2_y1"-1; 0016 END_IF 0017 ELSE IF "count_m2_y1"="count_m2_y1"+100 THEN 0018 "stop_m2"=1; 0019 "s_v" := "count_m2_y1"; IF "stop_sensor_z"=0 THEN GOTO 1 ELSE IF "moved"=1 THEN 0020 IF "count_moved"="count_moved"-63 THEN "stop_m2"=1; "moved"=0; 0021 ELSE "rs4"=1; "count_moved"-1; END_IF 0022 ELSE IF "count_moved"="count_moved"+63 THEN "stop_m2"=1; "moved"=1; 0023 ELSE "rs4"=1; "count_moved"+1; END_IF 0024 ELSE "rs4"=1; "count_m2_y1"-1; 0025 ELSE "rs3"=1; "count_m2_y1"+1; 0026 END_IF 0027 END_IF 0028 ELSE IF "s_v"-"s_q">0 THEN 0029 IF "count_m2_y1"="count_m2_y1"-200 THEN 0030 "stop_m2"=1; 0031 "s_v" := "count_m2_y1"; IF "stop_sensor_z"=0 THEN GOTO 1 ELSE IF "moved"=1 THEN 0032 IF "count_moved"="count_moved"-63 THEN "stop_m2"=1; "moved"=0; 0033 ELSE "rs4"=1; "count_moved"-1; END_IF 0034 ELSE IF "count_moved"="count_moved"+63 THEN "stop_m2"=1; "moved"=1; 0035 ELSE "rs4"=1; "count_moved"+1; END_IF 0036 ELSE "rs4"=1; "count_m2_y1"-1; 0037 ELSE "rs4"=1; "count_m2_y1"-1; 0038 END_IF 0039 ELSE IF "count_m2_y1"="count_m2_y1"+200 THEN 0040 "stop_m2"=1; 0041 "s_v" := "count_m2_y1"; IF "stop_sensor_z"=0 THEN GOTO 1 ELSE IF "moved"=1 THEN 0042 IF "count_moved"="count_moved"-63 THEN "stop_m2"=1; "moved"=0; 0043 ELSE "rs4"=1; "count_moved"-1; END_IF 0044 ELSE IF "count_moved"="count_moved"+63 THEN "stop_m2"=1; "moved"=1; 0045 ELSE "rs4"=1; "count_moved"+1; END_IF 0046 ELSE "rs4"=1; "count_m2_y1"-1; 0047 ELSE "rs3"=1; "count_m2_y1"+1; 0048 END_IF 0049 END_IF </pre>									
Symbol	Address	Type	Comment						
"count_m2_y1"	%MW10	Int							
"count_moved"	%MW14	Int							
"moved"	%M14.1	Bool							
"rs3"	%M0.3	Bool							
"rs4"	%M0.4	Bool							
"s_q"	%MW12	Int							
"s_v"	%MWB	Int							
"stop_m2"	%M14.0	Bool							
"stop_sensor_z"	%I.4	Bool							

Рисунок 59 – SCL код для программного блока «y7»

LAD программа для оси выдвижения (рисунок 60) имеет один программный блок SCL (рисунок 61) и две логических битовых кнопки. Первая, отвечающая за выполнение алгоритма, если стенд находится в запущенном режиме, за это отвечает бит разрешения, привязанный к программной булевой переменной «rs_main», и вторая, отвечающая за выполнения алгоритма выдвижения, если вертикальный привод находится в режиме остановки. привязанный к программной булевой переменной «stop_m2».

В строковой программе блока «z1» в теле цикла «IF» используется цикл «WHILE», который привязан к булевой переменной концевого выключателя «stop_sensor_z». Основную функцию в данной программе выполняют переменные «rs5» и «rs6» которые коммутируют RS триггеры из силовой подсети «MAIN» с рисунка 35, которые в свою очередь коммутируют реле.

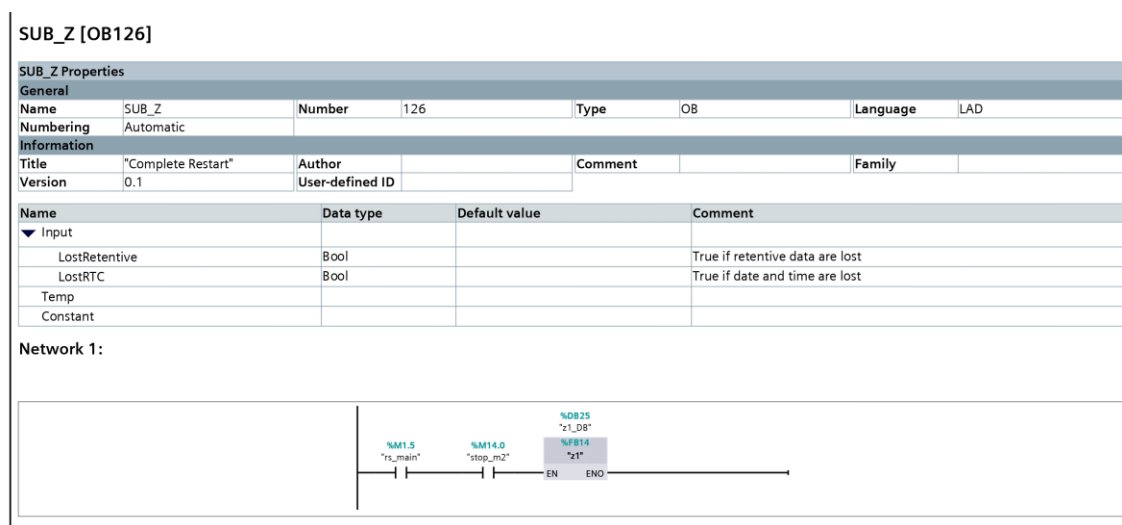


Рисунок 60 – Программа движения по оси выдвижения на языке LAD

z1 [FB14]

z1 Properties

General

Name	z1	Number	14	Type	FB	Language	SCL
Numbering	Automatic						

Information

Title	Author	Comment	Family
Version	User-defined ID		

Name	Data type	Default value	Retain	Accessible from HMI/OPC UA	Write-Visible in HMI engineering	Setpoint	Supervision	Comment
Input								
Output								
InOut								
Static								
Temp								
Constant								

```

0001 IF "stop_sensor_z"=1 THEN
0002   WHILE "stop_sensor_z"=0 DO
0003     "rs6"=1
0004   END_WHILE;
0005
0006 ELSE
0007   WHILE "stop_sensor_z"=1 DO
0008     "rs5"=1
0009   END_WHILE;
0010
0011 END_IF;
0012

```

Symbol	Address	Type	Comment
"rs5"	%M0.5	Bool	
"rs6"	%M1.6	Bool	
"stop_sensor_z"	%I1.4	Bool	

Рисунок 61 – SCL код для программного блока «z1»

Проект в среде программирования TiaPortal объединяет в себе все описанные LAD и SCL программы, а также блоки данных для подсистем, а также SCL блоков.

В данном разделе был произведен аппаратный и программный пусконаладочный процесс, с уточнением всех особенностей сопряжения ПК и ПЛК, блоков питания и релейной развязки. Написание алгоритма, а в дальнейшем и программы, было начато с аналитического алгоритма работы, который был переформатирован в блок-схемы с введением переменных и констант, и в завершении написания программы каждая блок схема приобрела вид LAD программы с блоками данных, а также функциональных блоков с SCL программой, которая состоит из ветвящихся циклов. Главное преимущество данного представления заключается в том что, при изучении блок-схемы и программы одновременно, а также опираясь на аналитический алгоритм, оператор станда, или изучающий его студент, с легкостью поймет каждую происходящую функцию и сможет опираясь на данные примеры, писать уже свои алгоритмы и программы.

3. Лабораторные работы

3.1 Лабораторная работа №1 «Ручное управление стандом»

1 Цель работы

Цель работы – приобрести навыки программирования в среде TiaPortal и научиться управлять стандом в ручном управлении.

2 Программа работы

2.1 Провести сопряжения ПЛК Siemens и TiaPortal.

2.2 Провести переименование стандартных тэгов в название привязанных к ним кнопок и реле.

2.3 Создать блок «Startup» и выбрать рабочую подсеть «Network 1».

2.4 В рабочий подсети «Network 1» разработать с помощью битовой логики программу из задания выданного варианта.

2.5 Составить блок схему разработанной программы соответственно выданному варианту.

2.6 С помощью встроенной экспортной функции извлечь блок программы в формат Pdf .

2.7 Подготовить и защитить отчет по работе.

3 Указания к выполнению работы

К пункту 2.2 – переименование тэгов выводов происходит в вкладке слева «Device configuration», двойной клик левой кнопкой мыши по выбранному выводу. Чтобы узнать к какому адресу подключена та или иная кнопка, следует воспользоваться индицирующей лампочкой у соответствующего вывода при замыкании. Чтобы узнать к какому адресу подключено то или иное реле, следует пронаблюдать гнездо подключения кабеля на винтовое соединение.

К пункту 2.3 –блоки операций создаются в вкладке «Program block» далее «Add new block».

К пункту 2.6 –экспорт документа производится выбором программного блока, и последующим щелчком правой кнопкой мыши на пункт «Print»

Таблица 3 – Варианты для бригад

Вариант	Задание
1	Кнопки №1-2 отвечают за привод М1 Кнопки №3-4 отвечают за привод М2 Кнопки №5-6 отвечают за привод М3
2	Кнопки №3-4 отвечают за привод М1 Кнопки №5-6 отвечают за привод М2 Кнопки №1-2 отвечают за привод М3
3	Кнопки №5-6 отвечают за привод М1 Кнопки №1-2 отвечают за привод М2 Кнопки №3-4 отвечают за привод М3
4	Кнопки №1-3 отвечают за привод М1 Кнопки №2-4 отвечают за привод М2 Кнопки №5-6 отвечают за привод М3

3.2 Лабораторная работа №2 «Работа с высокоскоростным счетчиком станда»

1 Цель работы

Цель работы – приобрести навыки программирования в среде TiaPortal, научиться подключать аппаратную функцию «High Speed Counter» (HSC) и считать импульсы энкодера.

2 Программа работы

2.1 Провести сопряжения ПЛК Siemens и TiaPortal.

2.2 Создать блок данных для переменных «DataBlock_1» и внести в него переменную «Count» и константы «Score_x» и «Score_y»..

2.3 Используя созданный в первой лабораторной работе блок «Startup» и рабочую подсеть «Network 1» добавить из стандартной библиотеки программный блок «HSC» и произвести его настройку.

2.4 Начать работу в блоке «Startup», перейдя к вкладке «Network 2», добавить: программный блок «HSC», блок «Move_to» (указав ячейку памяти «Count»), блок «Add» (указав ячейку памяти «Count» и слагаемое равное «1»).

2.5 Произвести замер движения в импульсах: одного оборота, размаха движения, импульсного расстояния между положениями расстановки на «складе» станда согласно варианту.

2.6 С помощью встроенной экспортной функции извлечь блок программы в формат Pdf .

2.7 Подготовить и защитить отчет по работе.

3 Указания к выполнению работы

К пункту 2.2 – при создании в блоке памяти переменной счета «count», следует задуматься об обнулении данных. Либо программе будет обнулять данные по завершении цикла («Retain» включен), в этом случае следует сразу фиксировать данные, либо оператор программы будет обнулять каждый цикл ячейку памяти вручную фиксируя данные («Retain» отключен)

К пункту 2.3 – аппаратное назначение входа под функциональное назначение энкодера находится во вкладке «General», «High speed counter (HSC)», «Hardware inputs».

При настройке блока HSC следует описать аппаратный идентификатор счетчика в блоке в графу «HSC». (по умолчанию номер идентификатора находится во вкладке «Hardware indentifier» и имеет номер «257»)

Таблица 4 – Варианты для бригад

Вариант	Задание
1	Произвести замер по вертикали от уровня ячеек «1» до уровня ячеек «3» и от уровня ячеек «2» до уровня ячеек «3».
2	Произвести замер и просуммировать данные движения по «прямоугольнику» (движение по номеру полок 1-2-4-6-5-3-1).
3	Произвести замер и просуммировать данные движения по «прямоугольнику с диагоналями» (движение по номеру полок 1-2-3-4-5-6-3-2).
4	Произвести замер и просуммировать данные движения по «песочным часам» (движение по номеру полок 1-2-5-6-1).

3.3 Лабораторная работа №3 «Автоматизированное движение»

1 Цель работы

Цель работы – приобрести навыки программирования в среде TiaPortal, научиться вводить SCL циклы и автоматизировать работу станда.

2 Программа работы

2.1 Провести сопряжения ПЛК Siemens и TiaPortal.

2.2 Пользуясь наработками лабораторной работы №2 добавить подсеть «Network 3», в списке блоков добавить программный блок SCL.

2.3 Используя синтаксис языка Pascal, и встроенные кейсы циклов написать код сравнивающий значение переменной «count» с константой «score».

2.4 Добавить в подсеть «Network 3» блок «empty box», кликнув на него, привязать к SCL программе, и согласиться, нажав клавишу «Enter», с автоматическим созданием блока данных «DB_1» (или «DB_name») для новой подсистемы.

2.5 Добиться выполнения движения по заданным координатам соответствующим данным в переменных «score_x» и «score_y» для вертикальной и горизонтальной оси.

2.6 С помощью встроенной экспортной функции извлечь блок программы в формат Pdf .

2.7 Подготовить и защитить отчет по работе.

3 Указания к выполнению работы

К пункту 2.3 – для того чтобы описывать в программе переменные стоит добавить переменные в список тэгов «Tag_1». Успешно описанная переменная отображается в левой колонке с привязанным к ней программным или аппаратным адресом (к примеру «%MW1.0», «%I1.0»). Также можно обнаружить в процессе написания программы небольшие отличия от

стандартного синтаксиса языка Pascal, однако среда программирования TiaPortal автоматически помогает пользователю исправляя некоторые части программы (в Pascal цикл IF заканчивается строкой «end;», а в SCL программе пишется «END_IF», что является интуитивно понятным)

Таблица 5 – Варианты для бригад

Вариант	Задание
1	Автоматизированное движение от локации №1 до локации №4
2	Автоматизированное движение от локации №3 до локации №6
3	Автоматизированное движение от локации №5 до локации №2
4	Автоматизированное движение от локации №6 до локации №1

3.4 Лабораторная работа №4 «Создание циклического автоматизированного движения»

1 Цель работы

Цель работы – приобрести навыки программирования в среде TiaPortal, научиться создавать автоматизированные циклы.

2 Программа работы

2.1 Провести сопряжения ПЛК Siemens и TiaPortal.

2.2 В блоке переменных «Datablock_1» добавить переменную отвечающую за состояние цикла в режиме ожидания, а переменную выбранного положения.

2.3 Создать новую программу SCL с новыми переменными, таким образом, чтобы нажатие одной из кнопок положения присваивало дополнительной переменной значение для вычисления которой получает данные счета от энкодера и в случае выполнения или не выполнения коммутирует или не коммутирует требуемое реле.

2.4 Добавить новый «empty_box», привязав к нему новую программу, а также создав блок данных «DB_name».

2.5 Создать программно-силовую подсистему с триггерами на выходе которого находятся выходы, привязанные к реле, таким образом, чтобы переменные из вычислительных подблоков, а также подблоков условий переключали триггеры в положение «TRUE».

2.6 С помощью встроенной экспортной функции извлечь блок программы в формат Pdf .

2.7 Подготовить и защитить отчет по работе.

3 Указания к выполнению работы

К пункту 2.5 – триггеры в данном случае являются кнопками разрешения с запоминающим состоянием и возможностью программной коммутации требуемый выход, а также вставлять условия отключения с логической функцией «OR».

Каждую силовую коммутационную линию следует располагать от начала общей ведущей линии параллельно друг другу. Для того чтобы добавить параллельную линию в одной подсети следует нажать на иконку «стрелки с углом» на панели инструментов.

Для того чтобы зациклить процесс работы также следует к подсети, отвечающей за выбор вторичного места добавить условие работы, а именно добавить блок «EQUAL» (эквивалент, равенство) и выбрать сравниваемую переменную, тип данных подключается автоматически, и число, с которым она будет сравниваться, а именно требуемые координаты: для горизонтальной две («score_x1» и «score_x2») и для вертикальной («score_y1», «score_y2», «score_y3»)

Таблица 6 – Варианты для бригад

Вариант	Задание
1	Все приводы двигают манипулятор одновременно
2	Привод вертикальной оси начинает работы после привода горизонтальной
3	Привод горизонтальной оси начинает работы после привода вертикальной
4	Горизонтальный привод начинает движение только когда вертикальный прошел половину расстояния до точки назначения

3.5 Лабораторная работа №5 «Полно структурный проект TiaPortal»

1 Цель работы

Цель работы – приобрести навыки программирования в среде TiaPortal, и создать полно структурный проект с выдвижением манипулятора и условиями безопасности.

2 Программа работы

2.1 Провести сопряжения ПЛК Siemens и TiaPortal.

2.2 Используя наработанные программы из лабораторных работ №1-4 добавить: силовую линию в главную подсеть для функции выдвижения, подсеть условий на концевых выключателях для оси выдвижения, блок безопасности выдвижения (только при остановке привода вертикальной оси) .

2.3 Создать новую программу SCL используя функцию условия «WHILE» для работы с концевиками и привязать к «Empty box» с блоком данных.

2.4 Добавить отслеживание при включении и следование в «нулевое» положение. Добавить подсеть «Main» которая разрешает работу для всех подсистем при нажатии на клавишу «СТАРТ».

2.5 С помощью встроенной экспортной функции извлечь блок программы в формат Pdf, графически описать алгоритм работы в виде блок-схемы.

2.6 Подготовить и защитить отчет по работе.

3 Указания к выполнению работы

К пункту 2.2 – силовой подсетью в данной лабораторной работе называется подсеть содержащая триггеры с условия установки высокого и низкого значения на выходе который следует к реле.

Отслеживание нулевого положения происходит за счет концевых выключателей, расположенных со стороны привода электромотора. Выполнение условия нулевого положения заключается в замыкании всех трех концевых выключателей.

К пункту 2.4 – для реализации условия разрешения следует создать булевою переменную без функции удержания («Retain») которая принимала бы значение триггера высокого уровня сигнала которого устанавливалось бы при нажатии клавиши «СТАРТ» и низкого уровня при нажатии клавиши «СТОП».

В данном разделе представлен блок лабораторных работ, при последовательном выполнении которых, студенты получают комплексное и поэтапное обучение работы со стендом трех осевого манипулятора. Автоматизация в данных лабораторных работах начинается от ручного управления с клавишного блока, далее включаются программные счетчики для отслеживания положения, и первые шаги к автоматизации стенда. И основной ступенью является переход в данных лабораторных работах к циклической автоматизированной работе и полно структурности проекта.

Заключение

Выводя завершающие мысли на данную тему, можно отметить, что внедрение автоматизации является актуальным направлением развития технического образования. Несмотря на высокую стоимость существующих учебных станков и стендов от производителей ПЛК, на рынке существуют дешевые конструкторы, такие как Fischertechnik, которые могут быть использованы для обучения автоматизации микроэлектроники. Созданный стенд, со всеми своими особенностями конфигурациями, является достаточно простым в понимании своего функционала, но одновременно и требующий усердия в полном освоении всех возможностей, как самого ПЛК Siemens, так и его среды программирования, в частности битовой логики. С постепенным освоением такого процесса как автоматизация трех осевого манипулятора помогут справиться представленные лабораторные работы, которые увеличивают свою обширность захватываемых и применяемых знаний с каждой последующей лабораторной работой, основываясь друг на друге.

При начале и изучении данной автоматизированной системы, следует анализировать процесс работы используя, такие материалы, представленные в данной работе как: аналитический алгоритм работы, блок схему работы стенда и LAD программу с блоками данных и переменных. Объединяя эти материалы и соотнося то как блок, схема соотносится с битовой логикой, большинство студентов с легкостью откроют для себя новый раздел программирования LAD, опираясь на который далее при должном упорстве можно будет легко адаптироваться под любой современный логический контроллер использующий один из стандартных видов программирования автоматизированных систем.

Список используемой литературы

1. Абдуллин Э. Б. Автоматизация координатных измерений в машиностроении: Учебное пособие / Э. Б. Абдуллин. – СПб.: Лань, 2016. – 160 с.
2. Аполлонский С. М. Теоретические основы электротехники. Электромагнитное поле: Учебное пособие / С.М. Аполлонский. - СПб.: Лань, 2018. - 592 с.
3. Бабакина Н. А., Колесников М. П. Современная промышленная электроника / Под ред. проф. Шкодырева В. П. – СПб.: СПбГПУ, 2013. – 267 с.
4. Бергер Ганс Автоматизация посредством STEP 7 с использованием STL и SCL и программируемых контроллеров SIMATIC S7-300/400. – 2-е издание, переработанное изд. – Нюрнберг: Департамент техники автоматизации и приводов, 2003. – 776 с.
5. Бергер Ганс Автоматизация с помощью Программ STEP7 LAD и FBD. – 2-е издание, переработанное изд. – Нюрнберг: Департамент техники автоматизации и приводов, 2003. – 605 с.
6. Басков С. Н. Система программирования ПЛК CoDeSys, методические указания для выполнения практических работ. - Челябинск: южно-уральский государственный университет (национальный исследовательский университет), 2018. - 90 с.
7. Дастин, Э. Тестирование программного обеспечения. Внедрение, управление и автоматизация / Э. Дастин, Д. Рэшка, Д. Пол; Пер. с англ. М. Павлов. — М.: Лори, 2017. — 567 с.
8. Деменков, Н. П. Управление в технических системах: учебник / Н. П. Деменков, Е. А. Микрин. - Москва: издательство МГТУ им. Н.Э. Баумана, 2017. - 452, [4] с.: ил.

9. ГОСТ Р ИСО 1503-2014. Эргономика. Требования к пространственной ориентации и направлениям движения органов управления. Введ. 01.12.2015. – М.: Стандаринформ 2015. – 36 с.

10. ГОСТ Р 51136-2008. Стекла защитные многослойные. Общие технические условия. Введ. 1.06.2009. – М.: ТК 391 "Средства физической защиты и материалы для их изготовления" 2008. – 32 с.

11. ГОСТ 29124-91, п. 6.3. Клавиатура ввода данных. Введ. 1.10.1992. – М.: Министерство радиопромышленности СССР 1992. – 12 с.

12. Зубарев, Ю.М. Автоматизация координатных измерений в машиностроении: Учебное пособие. 2-е изд., пер. и доп. / Ю.М. Зубарев, С.В. Косаревский. — СПб.: Лань, 2016. — 160 с.

13. Иванов, А.А. Автоматизация технологических процессов и производств: Учебное пособие / А.А. Иванов. – М.: Форум, 2012. – 224 с.

14. Иванов, В. Э. Основы проектирования микросистем на базе SIEMENS LOGO! и S7- 200 / В.Э. Иванов, Г.К. Конопелько, - Хабаровск : Изд-во Тихоокеан. гос. ун-та, 2016.- 160 с.

15. Остроух А.В. Основы построения информационно-вычислительных сетей: методические указания / А.В. Остроух. – Электрон. текстовые дан. – Красноярск: Научно-инновационный центр, 2017. – 56 с.

16. Лебедев А. М. и др. Следящие электроприводы станков с ЧПУ /А. М. Лебедев, Р.Т. Орлова, А. В. Пальцев. – М.: Энергоатомиздат, 1988. – 223 с.

17. Левенталь, Л. Введение в микропроцессоры: Программное обеспечение, аппаратные средства, программирование / Л. Левенталь. - М.: Энергоатомиздат, 2012. - 15 с.

18. Рекус, Г. Г. Общая электротехника и основы промышленной электроники / Г.Г. Рекус. - М.: Высшая школа, 2019. - 656 с.

19. Сергеев, А. И. Программирование контроллеров систем автоматизации: учебное пособие / А.И. Сергеев, А. М. Черноусова, А. С. Русяев; Оренбургский гос. ун-т. – Оренбург: ОГУ, 2016. – 125 с.

20. Советы по началу работы SINAMICS V20 . Компактное руководство по эксплуатации: [Электронный ресурс]. URL: https://vk.com/doc341515662_622879679?hash=zC42XbH5OE5C0FmrEd7qkBzvKGTu1EiNct3GCHR66H&dl=DCA5qpTt7gLSzvZ2Suu4Hs5Vqoicxd2WjJFNvsXPXD (Дата обращения 9.06.2023)

21. Схиртладзе, А.Г. Автоматизация технологических процессов: Учебное пособие / А.Г. Схиртладзе, С.В. Бочкарев, А.Н. Лыков. — Ст. Оскол: ТНТ, 2016. — 524 с. (Дата обращения 9.06.2023)

22. Cook, B. M. "Design automation." Computer-Aided Design 21, no. 8 (October 1989): 535. [http://dx.doi.org/10.1016/0010-4485\(89\)90064-x](http://dx.doi.org/10.1016/0010-4485(89)90064-x). (Дата обращения 9.06.2023)

23. Farrell, Bradley. "The role of the human in an age of automation." APPEA Journal 58, no. 2 (2018): 545. <http://dx.doi.org/10.1071/aj17188>. (Дата обращения 9.06.2023)

24. Hans Berger Automating with SIMATIC S7-1500 Configuring Programming and Testing. https://vk.com/doc341515662_622879596?hash=3CS0e7sw4xZbujq4NjZ5017VItWbmdBvdfvPTHnvgnH&dl=kExpGykwoMGzvdCT1NGGjSGBG8kPL7wyWazjbmJ5FGs (Дата обращения 9.06.2023)

25. Hugh Jack Automating manufacturing systems with PLCs. - Version 5.0 изд. - Free Software Foundation, 2007. - 839 с.

26. Kusiak, Andrew, and Mehmet Aktan. "Automation in Engineering Design." IFAC Proceedings Volumes 31, no. 15 (June 1998): 217–222. [http://dx.doi.org/10.1016/s1474-6670\(17\)40556-8](http://dx.doi.org/10.1016/s1474-6670(17)40556-8). (Дата обращения 9.06.2023)

27. Shah, Ankit P., Prof Kalpesh N. Shah, and Prof Harsh B. Joshi. "Design

Automation of Shell." Indian Journal of Applied Research 4, no. 4 (October 1, 2011): 214–16. <http://dx.doi.org/10.15373/2249555x/apr2014/65>. (Дата обращения 9.06.2023)